AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

1 OF 8
AD
A038796

1 OF 8

AD A038796

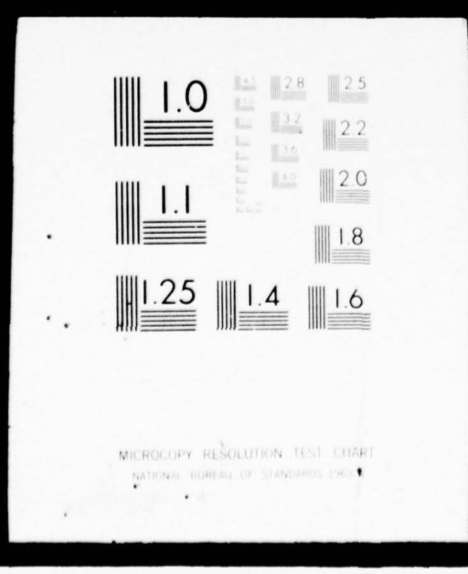


TABLE OF CONTENTS (CONTINUED)

2 4

B.S.

ADA038796

Technical Report: NAVTRAEQUIPCEN 73-C-0156-E003

MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAINING SIMULATOR (CATTS), DEVICE 16A3

Volume I - Sections 1 through 4

TRW Defense and Space Systems Group
One Space Park
Redondo Beach, California 90278

Contract N61339-73-C-0156
NAVTRAEQUIPCEN Task No. 3853

28 January 1977

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 19 NAVTRAEQUIPCEN 73-C-0156-E003-1 (18)	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) Mathematical Model User's Manual Combined Arms Tactical Training Simulator (CATTS), Device 16A3. Volume I. Sections 1 through 4.	5. TYPE OF REPORT & PERIOD COVERED Final Report, - 28 Jan 77
	6. PERFORMING ORG. REPORT NUMBER NAVTRAEQUIPCEN Task #3853
7. AUTHOR(s) (10) D.S./Adamson, E.C./Andreani, G.W./Archer, R./Cho J.H./Jensen; H.R. Milder; D.W. Myers; C.R. Peer; D.T. Unangst; W.H. Williams; J.A Wulfe	8. CONTRACT OR GRANT NUMBER(s) (15) N61339-73-C-0156 New
9. PERFORMING ORGANIZATION NAME AND ADDRESS TRW Defense and Space Systems Group One Space Park Redondo Beach, CA 90278	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Orlando, FL 32813	12. REPORT DATE (11) 28 January 1977
	13. NUMBER OF PAGES 1846
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 689p.	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Introduction, System Organization and Overview, Definition of Terms, Description of Data Tables

DISTRIBUTION STATEMENT A
Approved for public release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for Public

18. SUPPLEMENTARY NOTES

This Technical Report prepared by TRW Corp (Thompson, Ramos and Williams) of Redondo Beach, CA., as specified by NAVTRAEQUIPCEN Task #3853, Contract N61339-73-C-0156

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

CATTS (Combined Arms Tactical Training Simulator)
Systems Overview
CATTS Software (Background and Foreground)
Definition of Terms
Description of Data Tables

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

409 637

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This user's manual describes the mathematical model contained in the Combined Arms Tactical Training Simulator (CATTS) from the viewpoint of providing the user with the necessary insight into the structure and performance of the CATTS mathematical model in support of its operational use and maintenance. CATTS is a computer-supported, two-sided training simulator that is used to provide effective training for battalion field commanders and staff officers by realistically simulating ground combat operations between red and blue forces. The CATTS mathematical model is a large, detailed computer time-step simulation of the tactical battlefield environment, including detections, engagements, weapon firings, casualties, movement, and environmental effects for up to ninety-nine units. A complete understanding of the CATTS system operation, structure, performance and the use of the CATTS software can be constituted by using this manual together with the CATTS Operators Manual (NAVTRAEQUIPCEN 73-C-0156-E001), the Programming Report (NAVTRAEQUIPCEN 73-C-0156-A008), and the superindex program listings (NAVTRAEQUIPCEN 73-C-0156-A008).

2

Technical Report: NAVTRAEQUIPCEN 73-C-0156-E003-1

MATHEMATICAL MODEL USER'S MANUAL
COMBINED ARMS TACTICAL TRAINING
SIMULATOR (CATTS), DEVICE 16A3

Volume I - Sections 1 through 4

TRW Defense and Space Systems Group
One Space Park
Redondo Beach, California 90278
Contract N61339-73-C-0156
NAVTRAEQUIPCEN Task No. 3853

28 January 1977

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

NAVTRAEQUIPCEN 73-C-0156-E003

Prepared by:

D. S. Adamson

E. C. Andreani

G. W. Archer

R. Cho

J. H. Jensen

H. R. Milder

D. W. Myers

C. R. Peer

D. T. Unangst

W. H. Williams

V. A. Wulff

NAVTRAEQUIPCEN 73-C-0156-E003

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

1. INTRODUCTION

1.1 PURPOSE

The purpose of the User's Manual is to describe the mathematical model contained in the Combined Arms Tactical Training Simulator (CATTS) from the viewpoint of providing the user with the necessary insight into the structure and performance of the CATTS mathematical model in support of its operational use and maintenance. CATTS is a computer-supported two-sided training simulator that is used to provide effective training for battalion field commanders and staff officers by realistically simulating ground combat operations between red and blue forces. The CATTS mathematical model is a large, detailed computer time-step simulation of the tactical battlefield environment, including detections, engagements, weapon firings, casualties, movement, and environmental effects for up to ninety-nine units. This User's Manual, together with the CATTS Operator's Manual (delivered as NAVTRAEQUIPCEN 73-C-0156-E001), the Programming Report (delivered as NAVTRAEQUIPCEN 73-C-0156-A005), the superindex program listings (delivered as NAVTRAEQUIPCEN 73-C-0156-A008), and the Data Base Manual will constitute the basis for obtaining a complete understanding of the CATTS system operation, and the structure, performance, and use of the CATTS software.

1.2 ORGANIZATION

The User's Manual contains sufficient information to permit an understanding of how the CATTS mathematical model functions and will allow the user to maintain and support its operational use.

Section 2, System Organization and Overview, is an attempt to provide the user with an understanding of the context in which the CATTS mathematical model operates. It presents (a) the general physical organization of the CATTS system and how that physical organization helps accomplish the training

objectives of the CATTS system; (b) an overview of the CATTS computer system and computer/human interface elements and relates how those elements interact with user to accomplish the aims of the CATTS system; (c) and an overview of the CATTS software (mathematical model as well as interactive software) and the interrelationships of the software modules.

Section 3, Definition of Terms, defines 163 of the terms which characterize the major concepts, entities, and data tables used by the mathematical model. These term definitions will provide the user with an understanding of the simulation concepts intrinsic to the mathematical model. The term definitions are alphabetized and are subdivided into general, single-variable, and multiple-variable terms, with each of these containing an English-language and a programmatical definition for the benefit of the non-technical and technical user, respectively.

Section 4, Description of Data Tables, provides both English-language and programmatical descriptions of 27 of the CATTS mathematical model data tables. CATTS is largely a table-driven simulator, with many of the tactical decisions and outcomes calculated based on values found in input data tables. Some of these tables have complex structures, and there are often interactions between entities in several different tables. A knowledge and understanding of the structure, operation, and interaction of these tables is essential to the effective use of the CATTS system. Also included in this section are size limitations on the tables, and a cross-reference to the mathematical model modules which access and/or modify each table.

Section 5, Model Descriptions, describes the software routines pertaining to the mathematical model of CATTS. As was done in the case of the Trainer Programming Report, these routines have been organized into ten basic modules. Each module is described in the User's Manual using a format which presents: 1) a high-level description of the operation of the module and its submodules, including an abbreviated description of the subroutines and their principal inputs and outputs; 2) the modeling assumptions and data sources used; and 3) the important tactical and physical mathematical equations used. These module descriptions essentially reflect the configuration of the mathematical software in existence at the time that the CATTS system was delivered

to Fort Leavenworth, Kansas. No particular attempt was made to ensure that all modifications made to the CATTS mathematical model subsequent to delivery of the system to Fort Leavenworth were included in the model descriptions. The only exceptions are that the descriptions of the "firing from leading edge of unit" and "stand-off air" modifications are also included (see Sections 5.4, Ground Fire Module, and 5.8, Air Module, respectively).

Section 6, Examples and Applications, presents detailed examples of the use of the User's Manual. These examples, representing major possible applications of the manual are described in simple, step-by-step procedures.

Appendix A, Nomenclature, presents a brief description of the computer programs used to generate and update the CATTS nomenclature list, followed by an alphabetized listing of all the variables (input and output) that are used in the CATTS mathematical model. Included are definitions of the meaning of these variables, their units of measurement and allowable range of indexes, as well as the identification of the common blocks containing the variables.

2. SYSTEM ORGANIZATION AND OVERVIEW

2.1 PURPOSE OF CATTS SYSTEM

The Combined Arms Tactical Training Simulator (CATTS) was conceived as a solution to the problem of providing effective training for battalion field commanders and staff officers. In particular, the desire was for an automated training aid which would "approximate the decision-making experience which can now be obtained only through actual participation in combat operations".

In the CATTS system, the officers to be trained (also referred to as players or as trainees) operate from mockups of the battalion Tactical Operations Center (TOC), which contains fully functional government-provided radios and field telephones modified and connected to a sophisticated solid state communications system. The TOC also contains a simulated radio teletype (RATT), microphone monitor pickups, a public address system, and a multi-directional sound system. All these devices are monitored and controlled by a team of controllers (also referred to as instructors) operating from another room, appropriately referred to as the control room. See Figure 2-1 for the general arrangement of the CATTS system, and Figure 2-2 for an overview of trainee-controller communications. When a trainee uses one of the communications devices he communicates with a controller who plays the role of the person who would normally be at the particular communication net (i.e., higher, lower, adjacent commands, artillery, air support, etc.) Conversations anywhere within the TOC can be monitored by any controller. All voice communications are recorded on a twenty-track audo recorder. Each radio network may have static and/or jamming added at the discretion of the controllers. Directional battle and motor noises can be introduced by the controllers over the multi-directional speaker system. The simulated RATT communicates with alphanumeric display devices operated by the controllers.

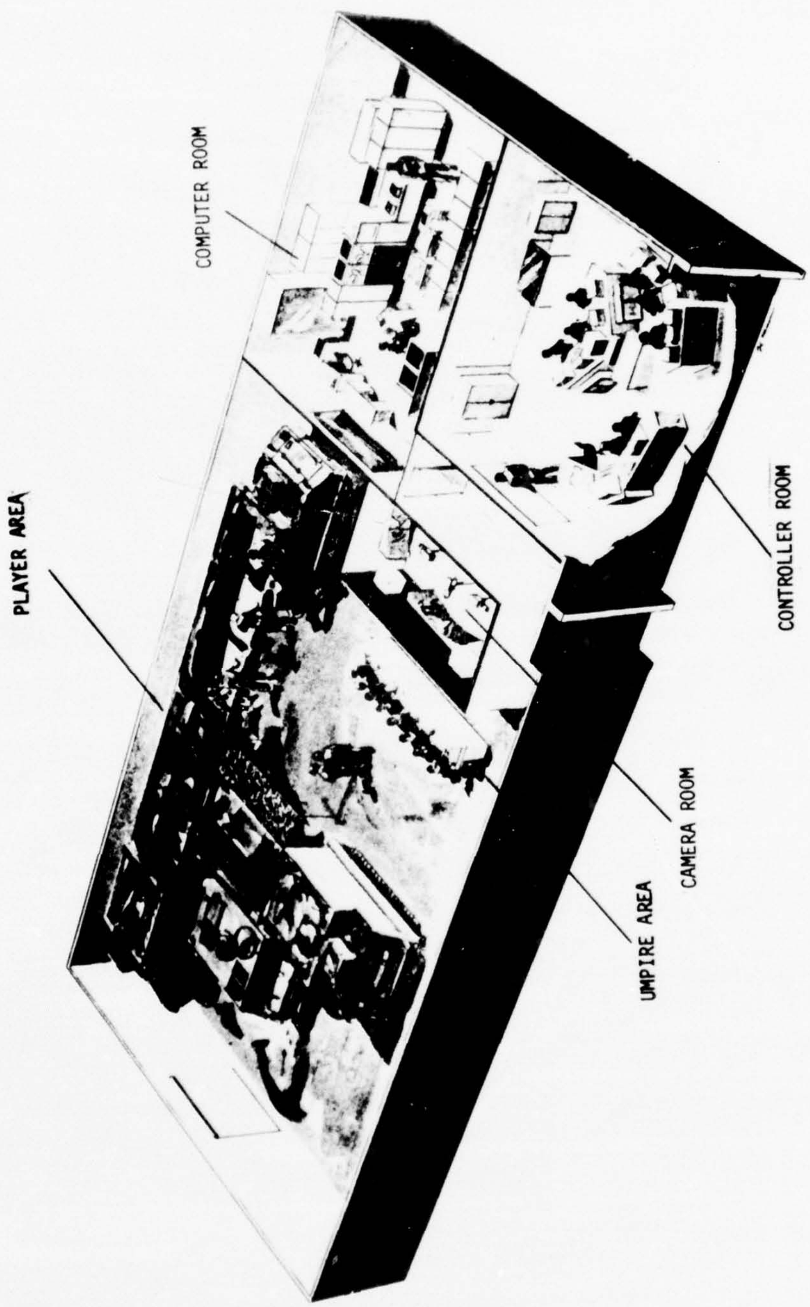


Figure 2-1. General Arrangement of the SATTS System

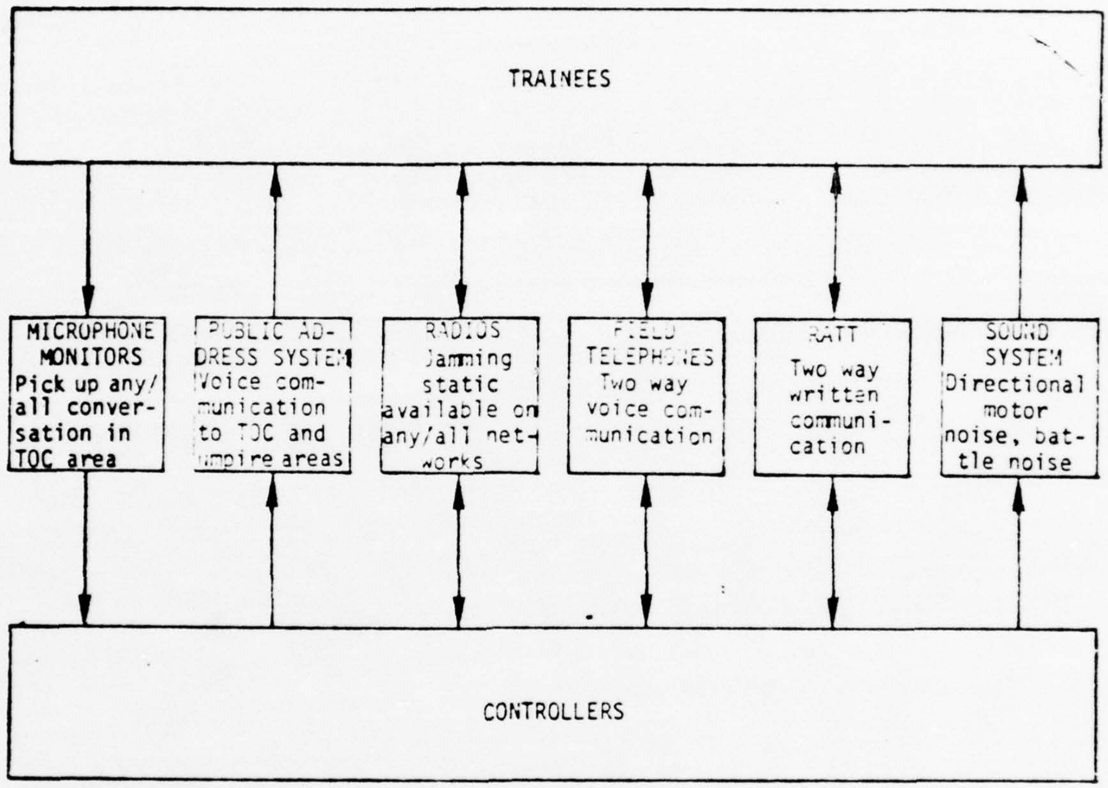


Figure 2-2. Overview of Trainee-Controller Communications

The training environment provided is a physically realistic one. The trainees operate in physically familiar surroundings using familiar communications equipment to communicate with the outside world. Their only source of information about the course of the battle comes over the communications system. From information provided by the controllers, the trainees must work as a team to maintain an accurate picture of what is going on, make command decisions under stress in real time, and communicate their decisions to the controllers playing the appropriate roles. (Note: CATTS can accommodate individual as well as team training.)

The technical problem the CATTS system attempts to solve is to give the controller an aid to calculate battle outcomes rapidly enough and realistically enough for training needs without constraining the freedom of action of the trainees. This is accomplished through the use of a large-scale computer system on which the battle is simulated by a mathematical model which calculates the battle outcomes, and a set of sophisticated interactive graphics programs and display devices which allow two-way communication between the controllers and the math model. An overview of this interactive capability is presented in Figure 2-3.

Thus we have a closed loop system in which the math model calculates outcomes and displays those outcomes to the controllers as alphanumeric messages on the alphanumeric display devices, and also as fully color military graphic symbology overlaying a full color military map on color television monitors. The appropriate role-playing controllers relay information to the trainees over the communications system. The trainees react and relay their orders and requests for support back to the appropriate controllers over the communications system. The controllers use graphic tablets and the color displays plus a complex set of command and control computer programs to enter the full spectrum of necessary military commands to the math model, which updates the necessary model variables to carry out the commands, thus changing all future battle outcome calculations. This closed loop, interactive system frees the controllers to dedicate their

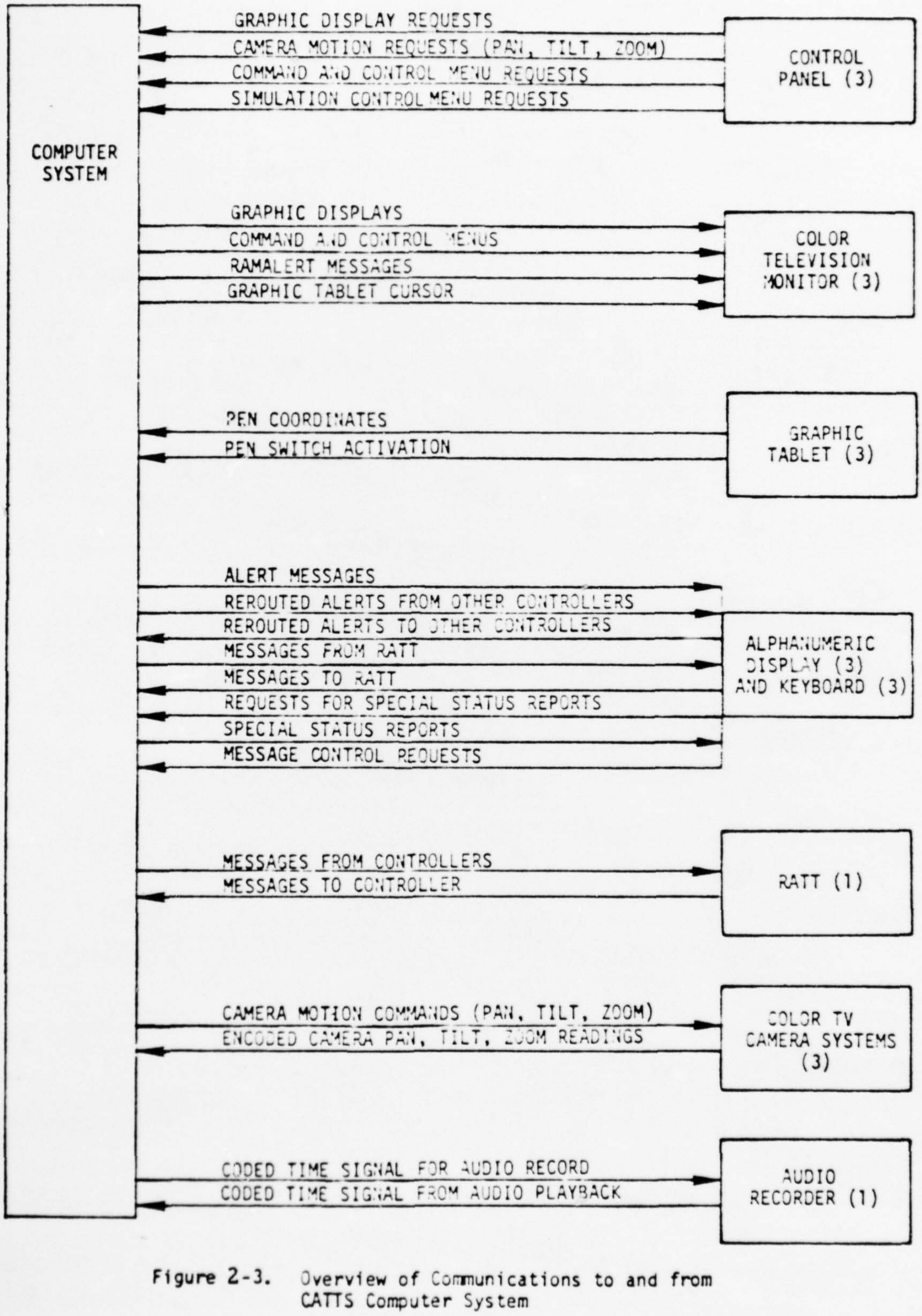


Figure 2-3. Overview of Communications to and from CATTS Computer System

efforts to role playing and to the training process, rather than to the calculation of casualties, movement rates, etc.

The basic CATTS system also includes an umpire or observer monitor area, which allows students, senior officers, or observers to monitor all aural communciations and all color graphics displays without actually participating in the exercise. In this way a group of students, for example, can watch, listen and learn from the mistakes of the trainees in the TOC.

A further system feature is the simulation control system, which allows the simulation to be frozen, replayed, restarted, or reinitialized at the command of the controllers. The simulation might be frozen during classroom break periods or for an admonitory warning over the public address system. A replay might be used to show trainees (gathered in the umpire area) what had "really" occurred and/or where they went wrong. A restart might then be used to reset the exercise to a point just before the fatal error was made, or to illustrate the outcome using a preferred tactic. Reinitialization to the same or a different scenario might be used to show that a tactic favorable under one set of conditions can be disastrous under another.

Many different kinds of diagnostic aids are available to assist the evaluation of trainee performance. Status reports are available throughout the game. Post-game processors produce a report showing the change in levels for every unit in the exercise, also a summary of red and blue casualties, and a sorted summary of all alphanumeric messages produced during the game. In addition, the replay of selected portions of the simulation with the appropriate color displays can be a valuable aid in reconstructing the events of the exercise. The audio communications can also be played back with or without the color graphics to aid in evaluating trainee communications technique or as an aid in the reconstruction of events.

2.2 SYSTEM OVERVIEW

An understanding of the CATTS software is difficult without some knowledge of the environment in which it operates. This section of this report first provides an overview of the CATTS hardware which has a software interface. Following this, the software overview is presented.

The CATTS system is housed in four separate rooms - the player room, the camera room, the control room, and the computer room, as shown in Figure 2-1. The only software interface to the player room is with the teletype, which serves as a simulated RATT. The camera room contains three color television cameras with software controlled pan, tilt, and zoom motors and encoders. The control room contains the audio recorder, the game clock, and three controller consoles. These consoles are the primary interactive software interface. The computer room contains a large-scale computer system plus the Ramtek color graphic display device.

Brief descriptions of the computer system and the various software interface devices will be presented in Sections 2.2.1 and 2.2.2. It should be emphasized that these descriptions are not intended to be a technically accurate nor exhaustive treatment of the devices, their operation, or their full capabilities. This information is presented solely as an aid to understanding the operation of the CATTS software.

An overview of this software is presented in Section 2.2.3. This overview is intended to provide a cursory software description. As such, it is provided as a way to give meaningful context to the detailed model description in Section 5. Many of the figures presented in Section 2.2.3 can only be completely understood after reading these detailed descriptions. The figures have been included here because even a partial understanding of them should aid the new initiate in understanding the CATTS system, and to provide a convenient set of reference figures for the individual who has more familiarity with CATTS.

2.2.1 Computer System

2.2.1.1 Computer Hardware

The computer chosen for CATTS was the Xerox Sigma 9 Model 3 with 128K (131,072 decimal) of 32 bit words in core storage. It is configured with three real-time clocks (which drive the command and control, graphic display, and map software, respectively), and alternate register sets (used to save interrupt processing time for the high volume of interrupts generated by the alphanumeric displays), The peripherals include a console teletype, a second teletype which serves as a simulated radio teletype (RATT), a card reader, a high-speed line printer, two magnetic tape drives, two high-speed line printer, two magnetic tape drives, two high-speed disk drives providing almost 90 megabytes of on-line storage, a character oriented communications (COC) device, and a digital input/output (DIO) device. The COC is the computer's hardware interface to the alphanumeric displays, and the DIO is the interface to all the switches, lamps, the game clock, the camera control unit, and other devices which require discrete inputs and/or outputs. An overview of the CATTS computer configuration is presented as Figure 2-4.

2.2.1.2 Operating System

The operating system used in the CATTS baseline is the Xerox Real-Time Batch Monitor (RBM) C04 version. Basically, this operating system allows simultaneous multiprogram operation in both real-time and batch modes. The interrupt-driven programs reside in an area of core called the foreground, while those running in the batch mode reside in the background. Execution automatically reverts to the current background program whenever all pending interrupts have been processed.

2.2.2 Computer/Human Interface Elements

The primary software interface in the CATTS system is to the three controller consoles in the control room. A single controller console is shown in Figure 2-5. Each console contains:

- Three communications systems panels, which have no software interface.

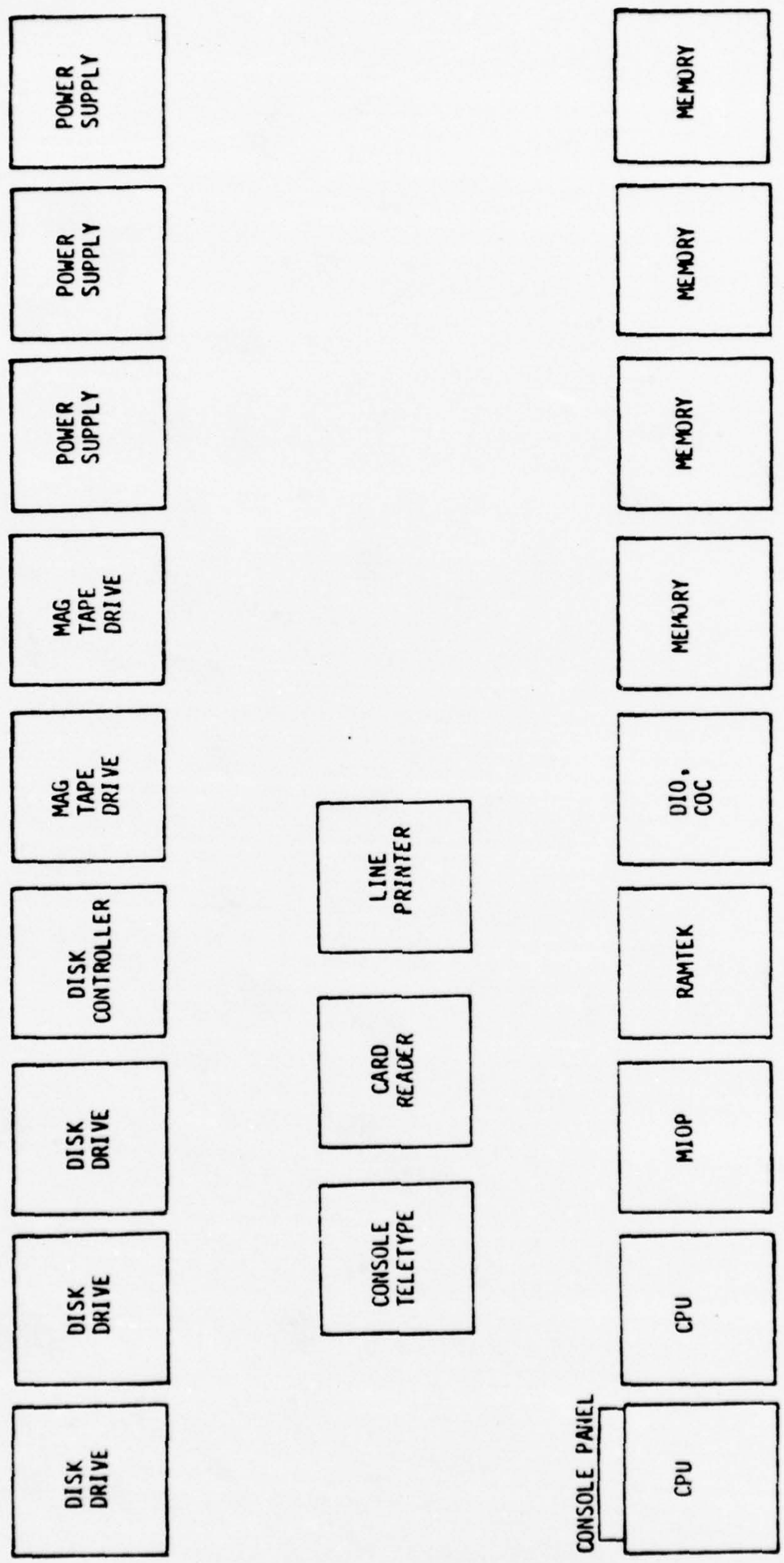


Figure 2-4. Layout of CATTS Computer Hardware

Figure 2-5. One of the Three CATTS Controller Consoles

- One color television monitor - the only software interface to this device is an indirect one, since the image on the monitor is composed of the map image plus the graphic display overlays, and both camera position and displays are software driven.
- A graphic tablet used to select menu items from menus displayed on the monitor.
- An alphanumeric display device with keyboard.
- A printer connected to the alphanumeric display, with no direct software interface.
- A control panel containing the switches used to request graphic displays, camera motion, and command and control menus, among other things.

The control room also contains:

- A game clock.
- An audio time encoder/decoder.

The camera room contains three color television camera systems, one for each controller console, each of which is comprised of:

- A large full-color military map mounted on a cylindrical map board, with no software interface.
- A color television camera with zoom lens, mounted on a pan/tilt table, with no software interface.
- A zoom motor and a zoom encoder.
- A pan motor and a pan encoder.
- A tilt motor and a tilt encoder.

The 9 motors are software driven through a single piece of interface hardware referred to as the Camera Motion Control device, also located in the camera room. The 9 encoders are read-only devices read by the software.

Figure 2-6 shows the general arrangement of the camera system. A more detailed description of the elements of the camera system is presented in Section 2.2.2.7.

The player room contains:

- A teletype serving as a simulated RATT.

The computer room contains:

- The Ramtek color graphic display device.

All of these devices and their interfaces will be described in slightly more detail in the remainder of this section.

2.2.2.1 Control Panel

The control panel serves many functions in CATTS. It is the method by which the controller requests camera motion, graphic displays, command and control, and simulation control, and by which he operates the audio recorder, continues the game after a freeze, and determines game status. The various switches and lamps on the panel are digitally coded, and these digital codes are passed from the panel to the CATTS software through the DIO, and vice versa. A control panel is shown in Figure 2-7. One is shown in place at a controller console in Figure 2-5.

The control panel is broken down into functional areas, shown in Figure 2-8. This panel contains the following switches:

- A set of alternate-action, backlighted push button switches used to request graphic display types.
- A set of alternate-action, backlighted push button switches used to specify force (red/blue) and level (platoon/company/battalion/higher) for requested command and control menus.
- A set of momentary push button switches used to request specific command and control menus. There is one switch for each menu type.
- A set of lamps used to indicate whether the system is running, frozen, or in replay.

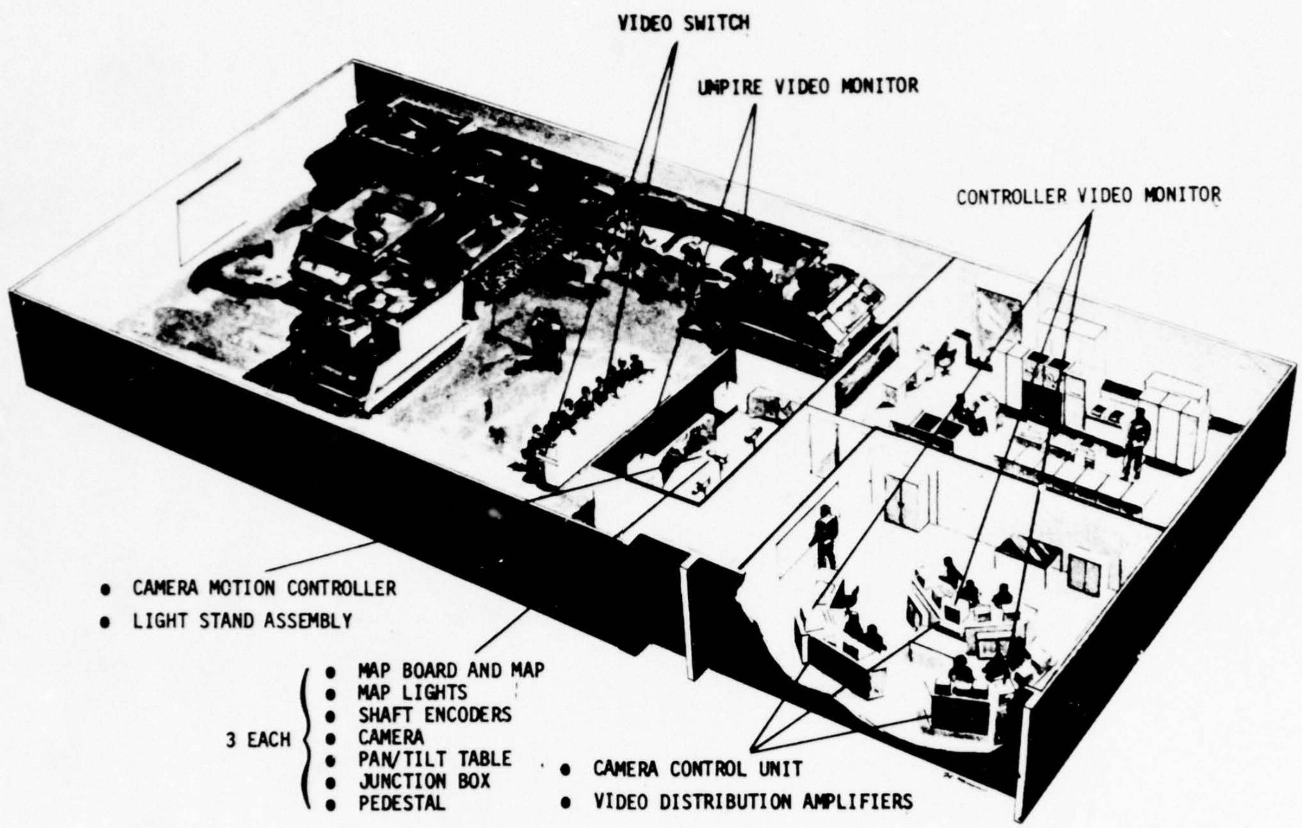


Figure 2-6. General Arrangement of the Map/Video Subsystem Hardware

Figure 2-7. Control Panel

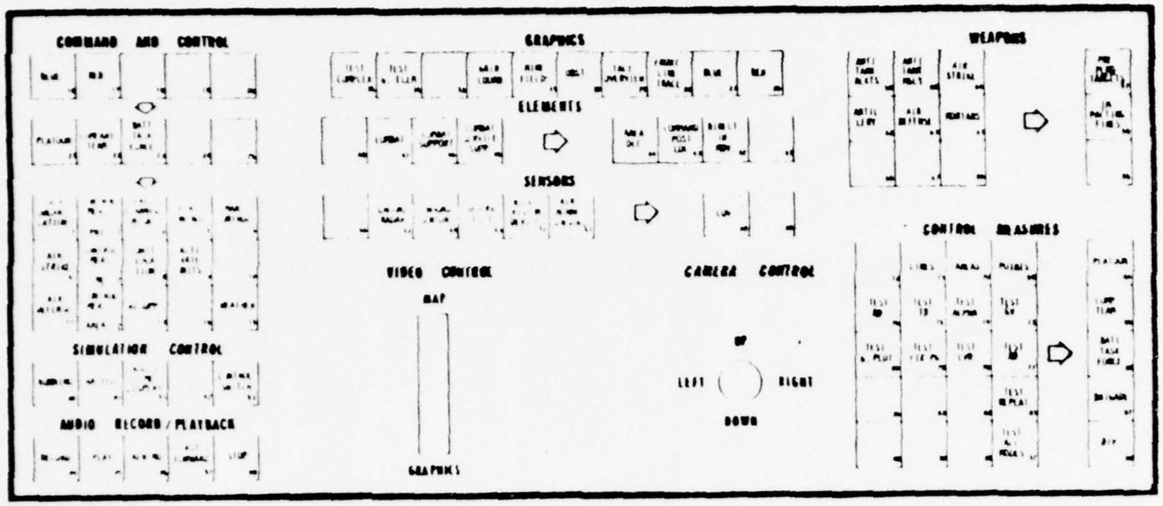


Figure 2-8. Identification of the Control Panel Switches and Lamps

- A six-position switch used to request camera movement - pan left or right, tilt up or down, zoom in or out.
- An alternate-action push button switch used to request the simulation control menu and to terminate a freeze, recommencing game exactly at the point the freeze began. This switch is active only at the principal controller console, which is designated by data base input.
- An alternate-action backlighted push button switch used to request audio time display.
- A set of momentary push button switches used to control the audio recorder. These switches are active only on console 1 and have no software interface.

2.2.2.2 Color Graphics Displays

Color graphic display on the CATTS system is accomplished through the use of the Ramtek display device. The Ramtek is a hardware device which converts digital inputs into analog color video signals. This device is connected to the computer system through the MIOP where it is treated like other peripheral devices such as the line printer. The Ramtek outputs are organized into 3 channels (one for each controller). Each channel has two subchannels - red and blue-green. By combination of these subchannels, black, red, blue-green, and white displays can be generated. The Ramtek has the capability of creating displays in several different modes. By use of these different modes, lines, military symbols, and alphanumeric data are easily generated. A picture of the Ramtek is shown in Figure 2-9.

The signals which emanate from the Ramtek are electronically added with those generated by the three video cameras. The resulting signals are displayed at the three controller stations on the video monitors. The relative intensity of Ramtek and camera video images is controlled by the knurled knobs on the control panels, shown in Figures 2-7 and 2-8 as the "video control".

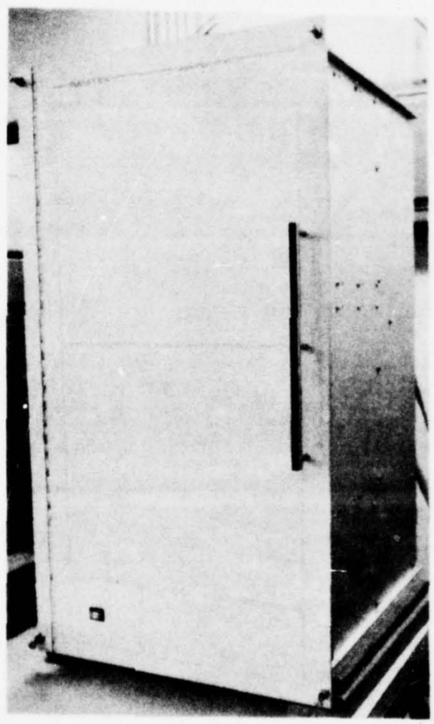


Figure 2-9. Ramtek Device

2.2.2.3 Alphanumeric Displays

The alphanumeric displays are standard CRT displays with keyboards, which are operated in CATTS in full duplex mode at 2400 baud through the COC. Each has an attached 300 baud matrix printer. These displays are the principal interface of the ALERT/RATT software. They are used primarily for display of alphanumeric messages generated by the CATTS math model whenever any of a large number of things (such as a unit's running low on fuel or ammunition) happens. They are also used for two-way communications between the controllers and the simulated RATT in the TOC, and for the display of special unit status reports requested by the controllers. One of the alphanumeric display devices is shown in Figure 2-10 with the attached hard copy device. An alphanumeric display as it appears when in use in CATTS is shown in Figure 2-11. The location of the alphanumeric displays in CATTS can be seen from Figure 2-12. One of the displays can also be seen in place at a controller console in Figure 2-5.

2.2.2.4 Radio Teletype (RATT)

The RATT is simulated in CATTS by the use of a standard Xerox supplied teletype station which is actually a part of the computer system. The location of the RATT is shown in Figure 2-13.

2.2.2.5 Game Clock

The CATTS game clock is a lighted digital (alphanumeric) display of the current five-digit (day, hours, minutes) game time. It is driven through the DIO, and is placed where it is visible to the controllers at all three consoles. The CATTS game clock is shown in Figure 2-14.

2.2.2.6 Audio Time Encoder/Decoder

The audio time encoder/decoder is a device which converts a digital game time code obtained through the DIO into a tone-coded audio signal which is recorded on a special channel of the twenty-channel audio tape recorder. On playback, it converts the audio signal back into a digital one which is passed through the DIO to the software. The audio recorder is in the left-most cabinet shown in Figure 2-15.

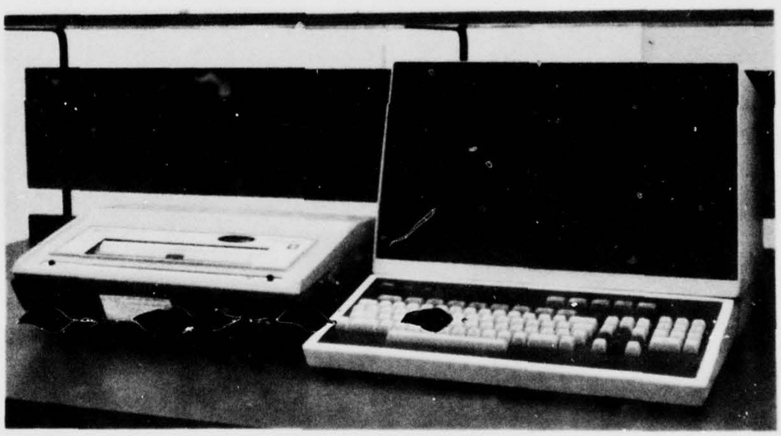


Figure 2-10. Alphanumeric Display and Keyboard. and Hard Copy Device

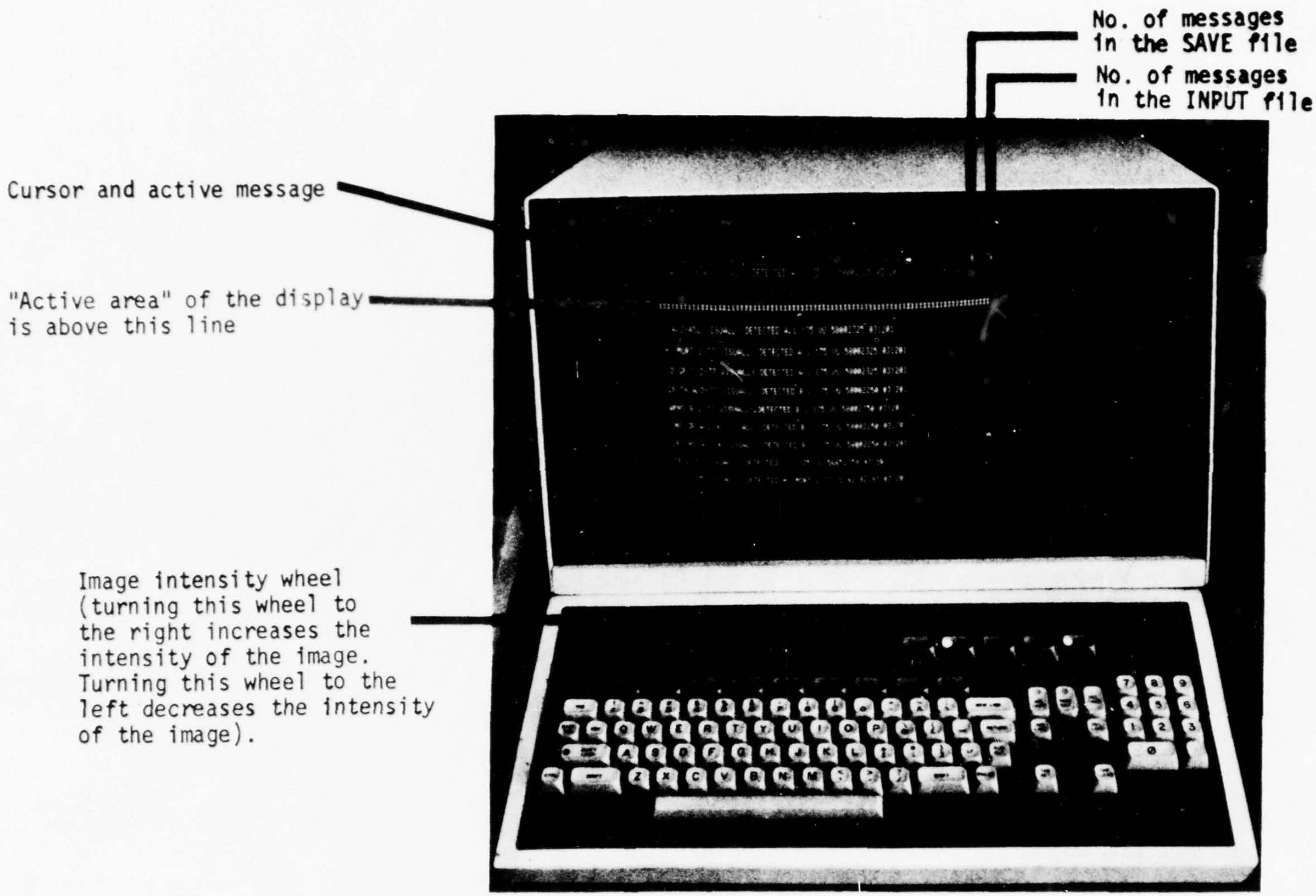


Figure 2-11. Organization of the Alphanumeric Display During Alerts Processing

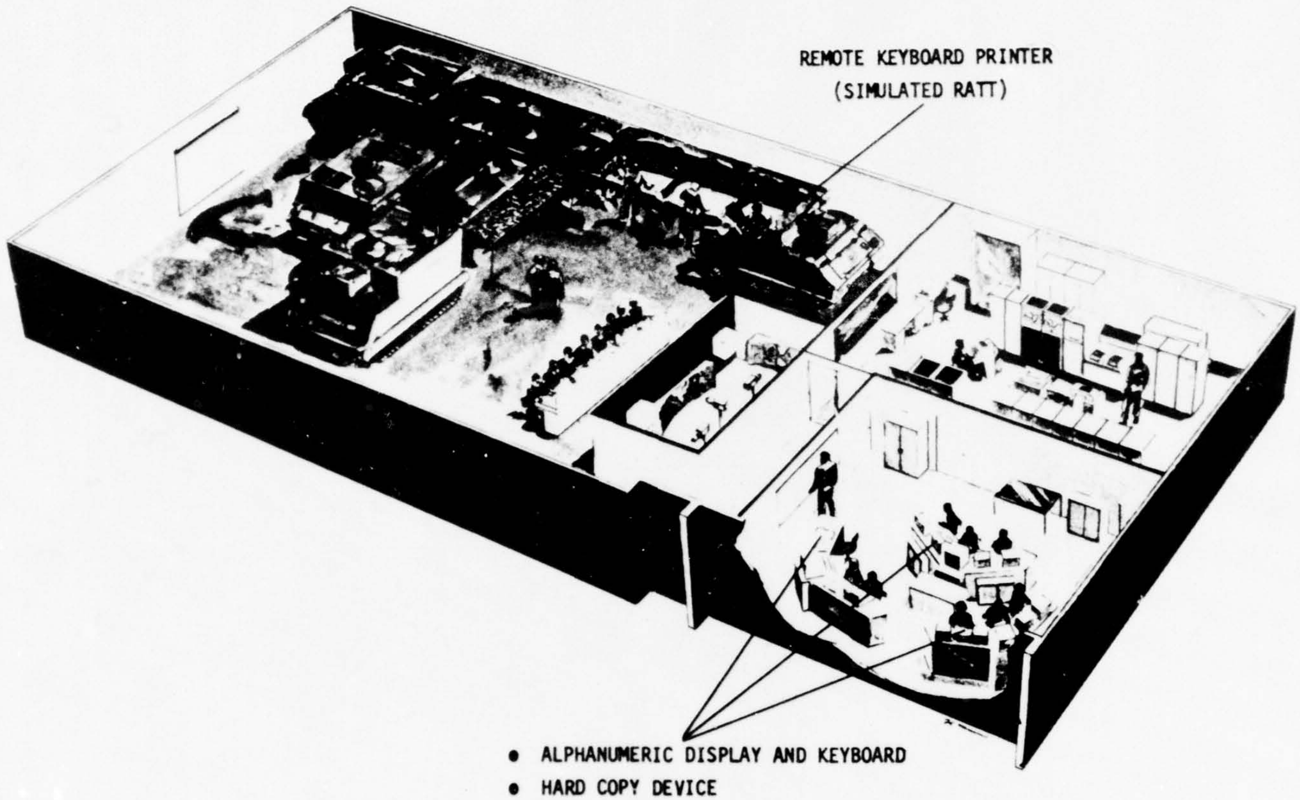


Figure 2-12. General Arrangement of the Alert/RATT subsystem Hardware

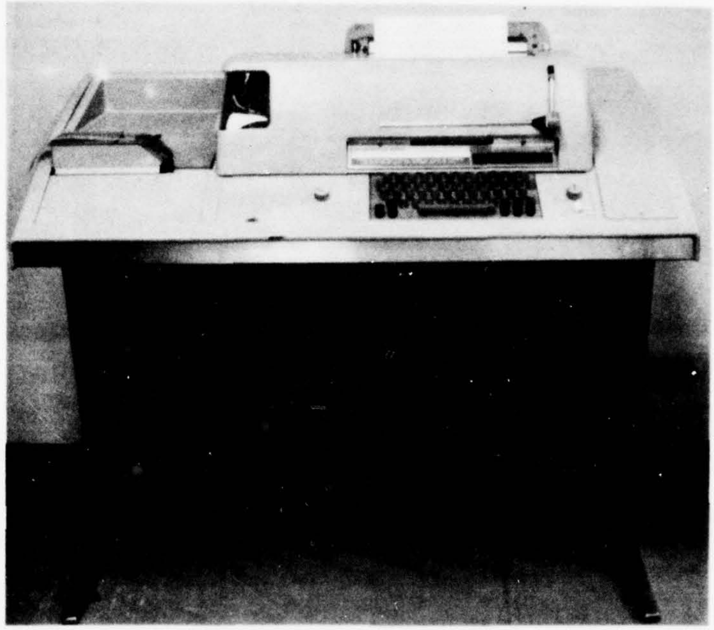


Figure 2-13. Communications Teletypewriter (Simulated RATT)

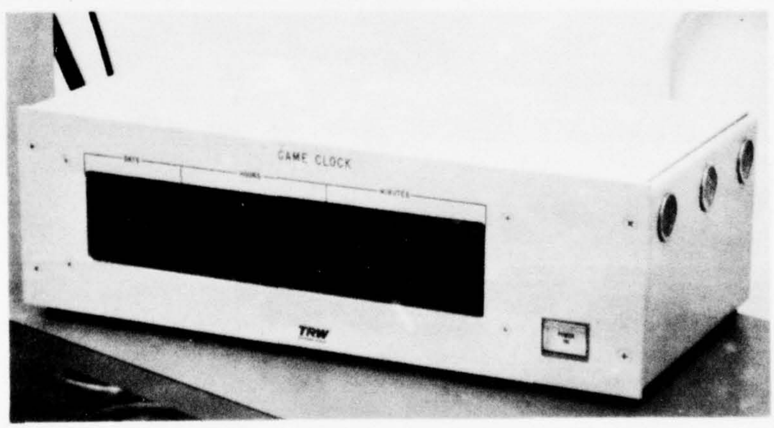


Figure 2-14. Game Clock (Not Operating at Time of Photograph)

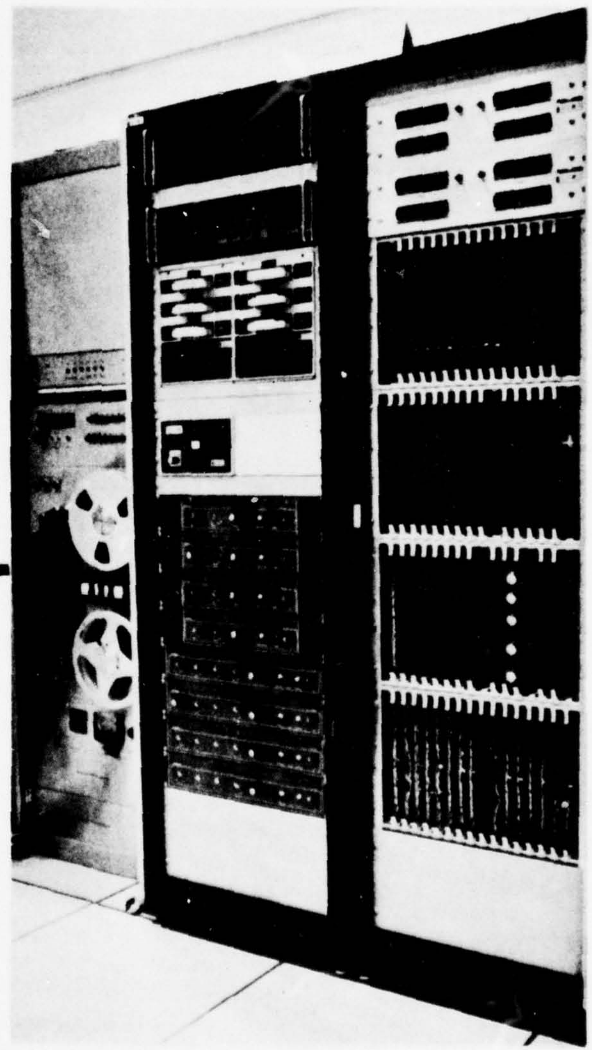


Figure 2-15. Communications Equipment Racks with Audio Recorder

2.2.2.7 Camera System

The CATTS map software drives the cameras by sending digitally coded signals through a special DIO module called a Driver Module, which turns relays controlling the pan, tilt, and zoom motors on and off. The pan, tilt, and zoom encoders on each camera convert the current pan, tilt, and zoom locations into three 13 bit digital codes which are transmitted through the DIO Input Module to the map software. The video image from each camera is cabled directly to a resistive mixing device in the corresponding control panel, where the signals are added with the graphics video signals from the Ramtek, amplified, and finally displayed on the television monitor. Figure 2-16 shows one of the three cameras used in CATTS. Figure 2-17 shows some of the other components of the camera system.

2.2.2.8 Graphic Tablet and Pen

The graphic tablet used in CATTS is a glass tablet with small microphones mounted along two edges, together with a ball-point pen which generates small electric sparks across a gap near the ball point. The lengths of time required for the sound of the spark to travel from the pen to the edges of the tablet are measured and converted by a control unit to an (x, y) coordinate pair. Figure 2-18 shows one of the tablets used in the CATTS system.

In CATTS, the graphic tablets are used primarily to make selections from the command and control menus. When the software detects that a controller has requested a command and control menu (by pressing one of the command and control momentary switches on the control panel), it creates the menu display and begins transmitting pen spark commands to the pen, through the DIO, at a rate of ten per second. The pen position coordinates are also read, through the DIO, ten times per second. Each time pen position is read, a cursor drawing instruction is sent to the Ramtek, which creates a cursor display in the location on the television screen which corresponds to the pen's location on the tablet. When the controller wishes to select a menu item, he first moves the pen to position the cursor over

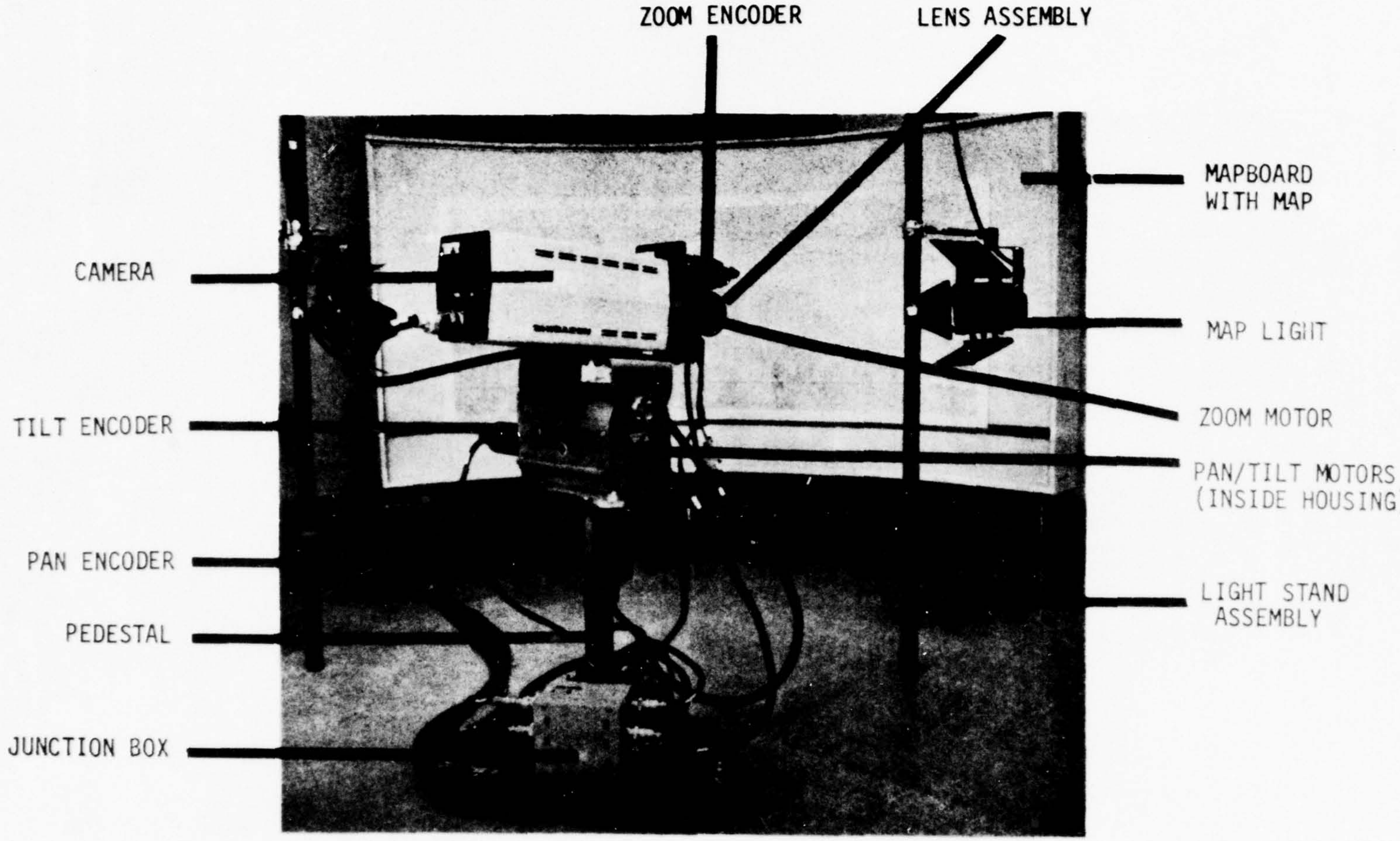
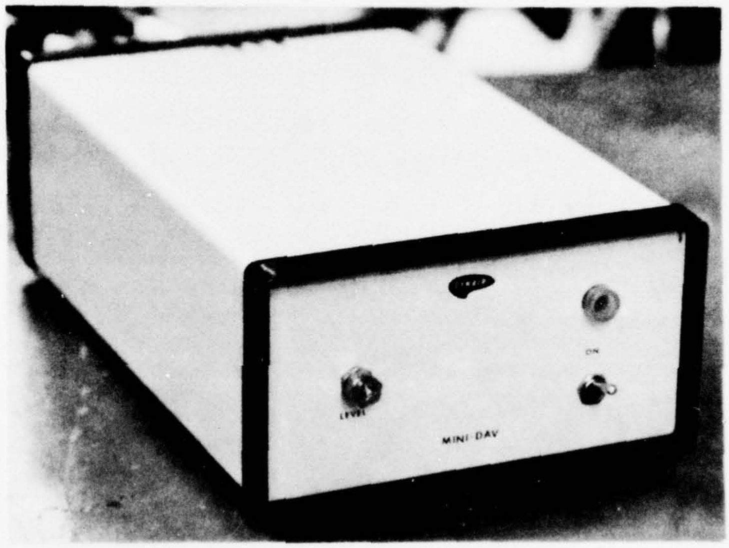


Figure 2-16. Map and Camera Hardware (This Equipment is Remotely Operated by the Controller)

A. Camera Control Unit



B. Video Distribution Amplifier

C. Camera Motion Controller

Figure 2-17. Video Hardware (This Equipment is Operated by Maintenance Personnel)

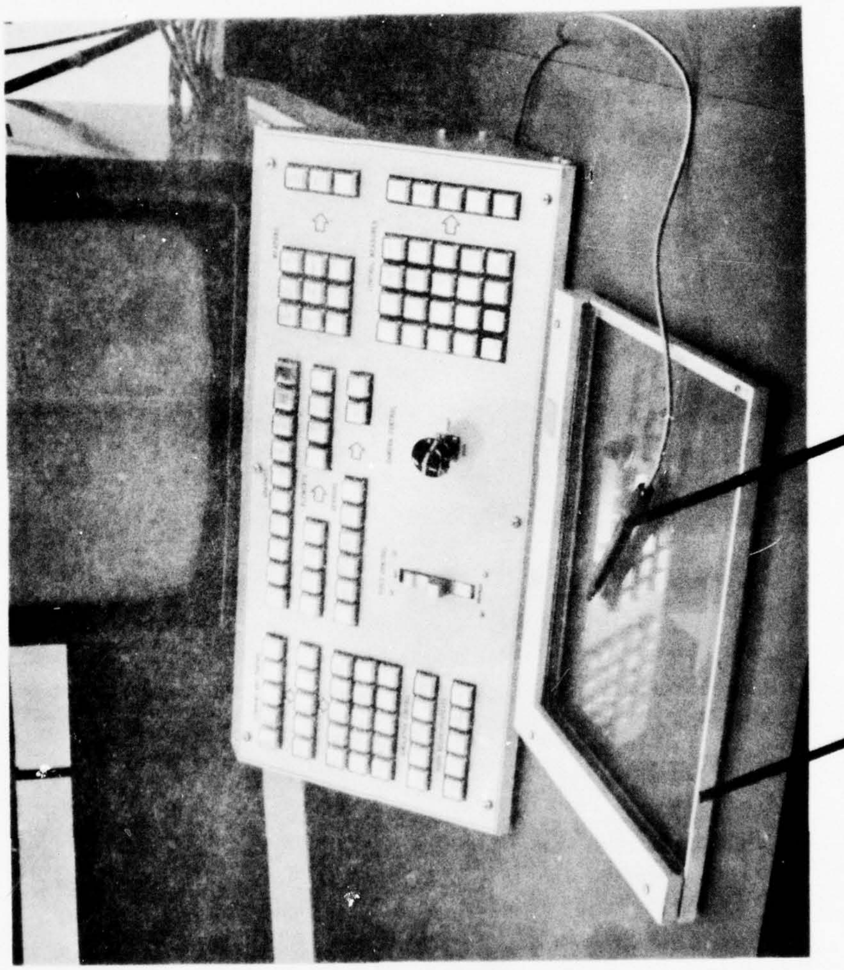


Figure 2-18. Graphic Tablet and Pen

that item on the television screen. Then he presses the pen down against the tablet. There is a special switch in the tip of the pen (called a tip-switch), and when the switch is activated a special signal is transmitted to the DIO, where it is read by the software. The software then generates the Ramtek commands necessary to change the menu display in accordance with the selection just made, and the process continues. Figure 2-19 demonstrates the use of the graphic tablet, while Figures 2-20 and 2-21 show details of menu selection. Figure 2-22 shows the control unit for one of the tablets.

2.2.3 CATTS Software

The CATTS software is divided into two types - small, fast, interactive programs which must have fast response to controller inputs, and the large CATTS mathematical model. The interactive programs run in the foreground mode and are principally concerned with communicating data and commands between controllers and the mathematical model. The mathematical model runs in the background mode, calculating battle outcomes, casualties, etc. Figure 2-23 shows an overview of the core usage of the individual programs.

2.2.3.1 Background Software

The CATTS mathematical model is a large, detailed, complex digital computer simulation of the tactical battlefield environment. It is a time-step model, with timesteps of one minute (except for air units, which have quarter-minute steps). It calculates, for each minute of battle, the detections, engagements, fires, casualties, movement, and environmental effects for up to ninety-nine units. The baseline scenarios have units which vary in strength from a squad to a battalion, with the normal level of platoon for friendly units and company for aggressor units.

Because the model is not interrupt-driven, it runs as a background program, and is often referred to as the "background software". Figure 2-24 is an overview of mathematical model communications with other system elements.

A. Center

B. Lower Right

Figure 2-19. Selecting a Location on the Screen Using the Graphic Tablet

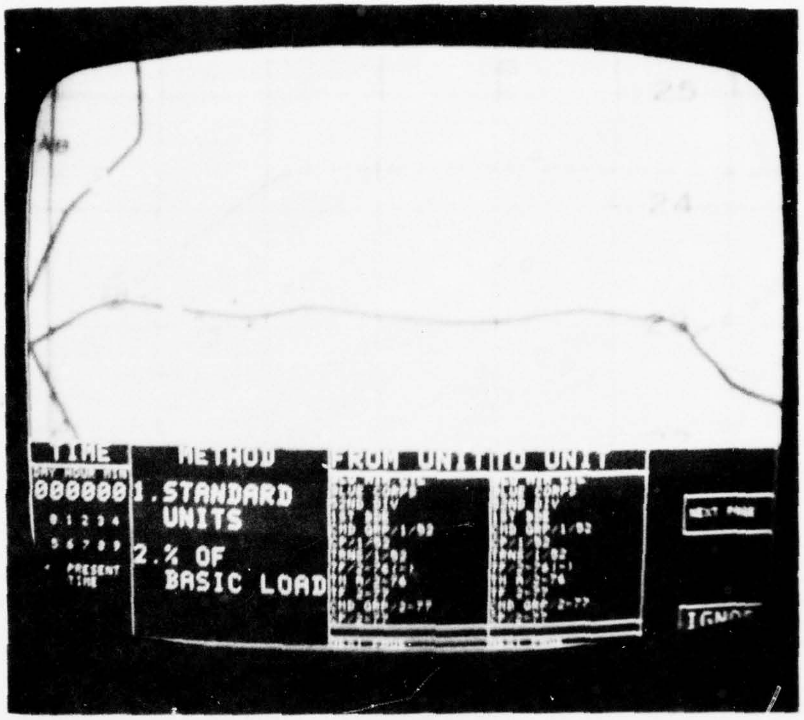


Figure 2-20. Resupply Menu Without Graf Pen Cursor

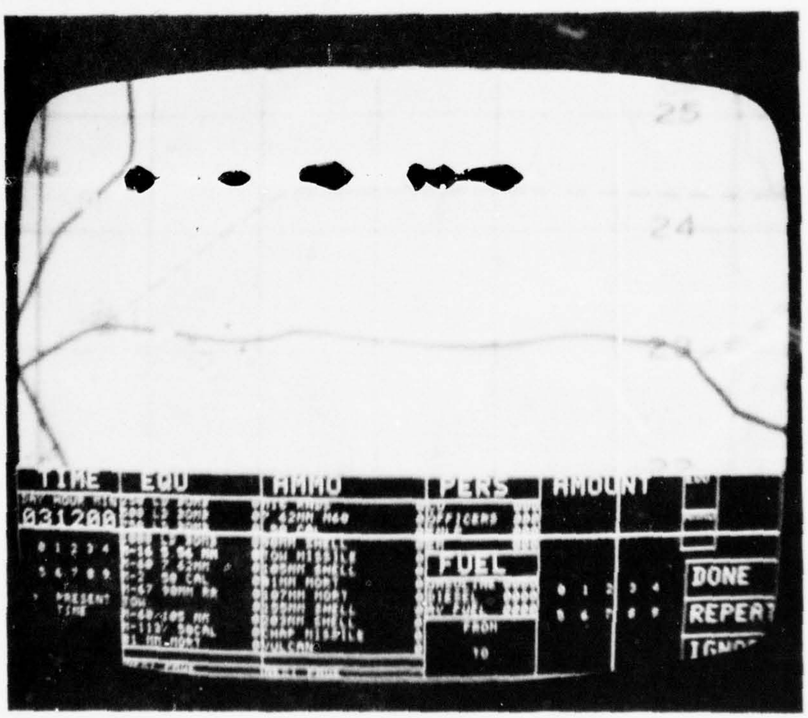


Figure 2-21. Selection of a Menu Item Using a Graf Pen

Figure 2-22. Graf Pen Controller (This Equipment is Operated by Maintenance Personnel)

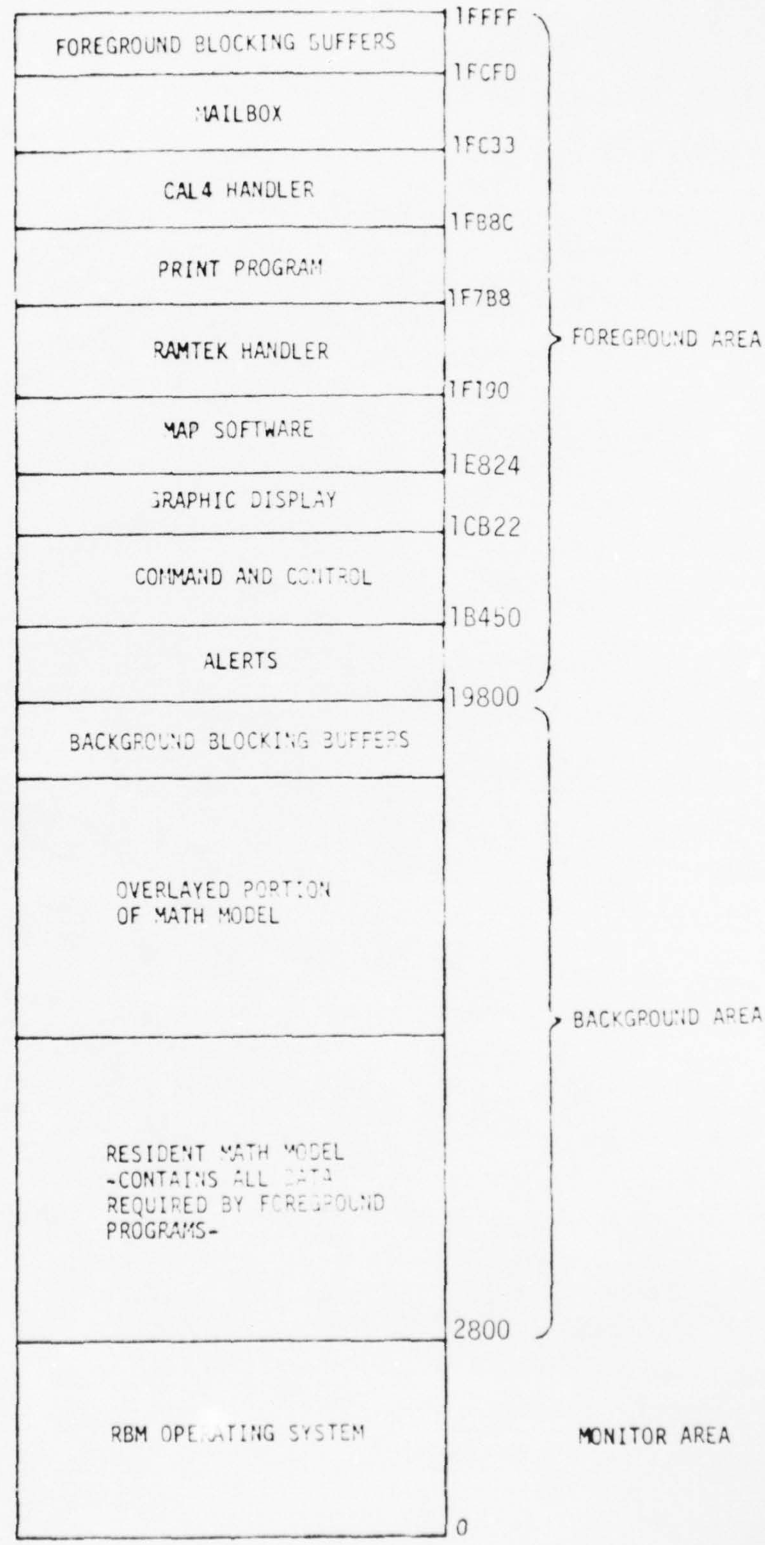


Figure 2-23. Computer Core Usage in CATTS

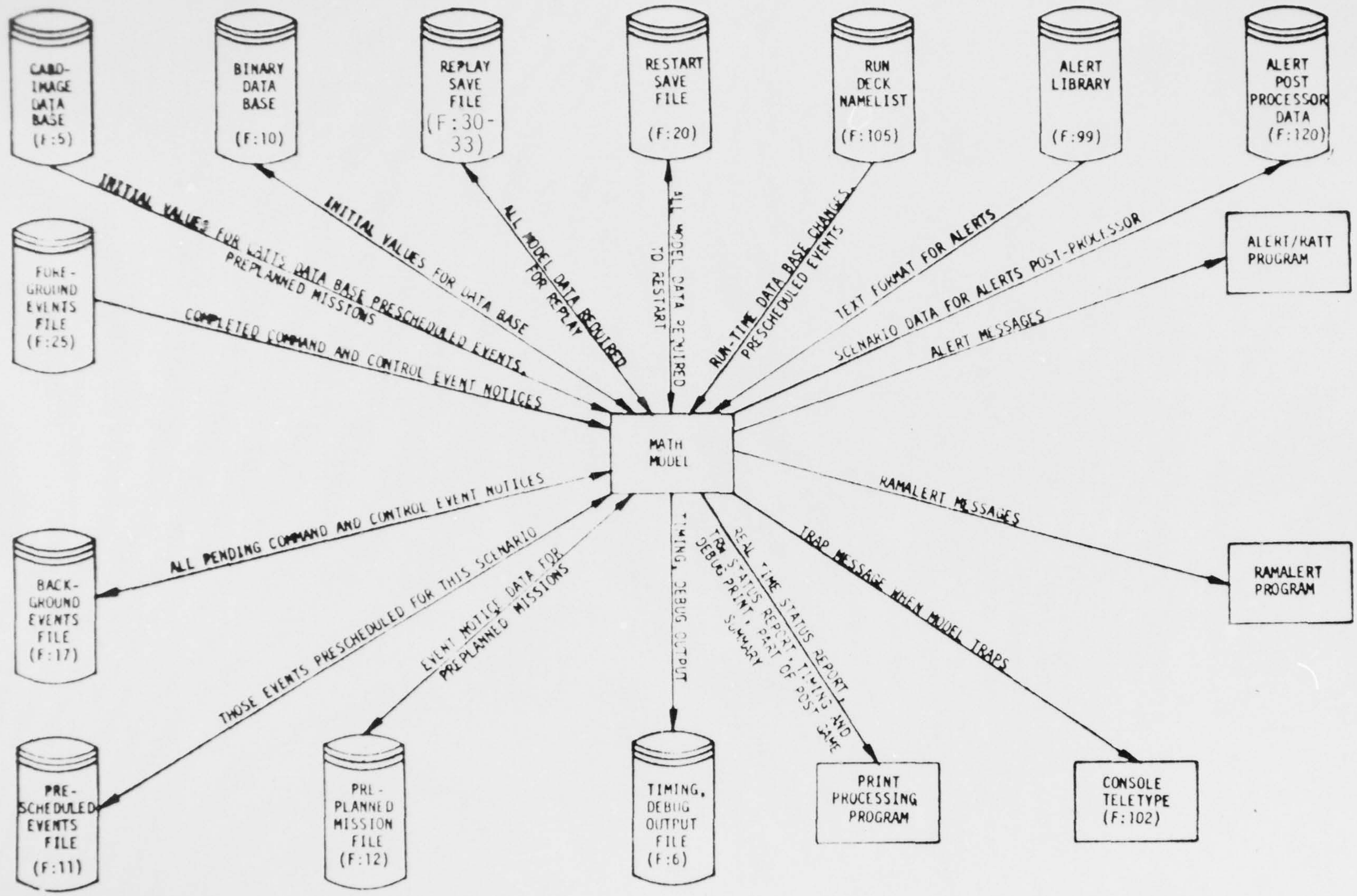


Figure 2-24. Overview of Math Model Information Flow
(Includes Xerox File Assignments)

The model is functionally divided into ten modules, each with a specific function. Each module is described in detail in Section 5 of this manual. A brief overview of each is presented as the remainder of this section.

2.2.3.1.1 Executive and Simulation Control Module

The Executive and Simulation Control Module has the responsibility of overseeing mathematical model execution. It moves the correct overlay segments to/from core when required, directs the execution of the various other modules, handles the interface with the foreground programs, saves the data necessary for replay and restart on disk files, and performs most of the functions of simulation control. Figure 2-25 shows the model overlay structure, while Figure 2-26 shows a logical flow diagram of the operation of this module. A more detailed description is given in Section 5.1.

2.2.3.1.2 Environmental Module

The Environmental Module has two purposes. One is to calculate the existance of lines of sight between eligible ground units, considering terrain relief and vegetation interaction. This is accomplished by a complex model using a large terrain data base developed from Defense Mapping Agency (DMA) provided data. A detailed description of the line of sight model is presented in Section 5.2.1.

The second purpose of ~~the~~ Environmental Module is to update the global weather conditions, which include:

- temperature
- relative humidity
- weather class (selected from 11)
- meterological visibility
- ambient light level
- wind velocity
- wind direction

This is accomplished by a weather model which is described in detail in Section 5.2.2.

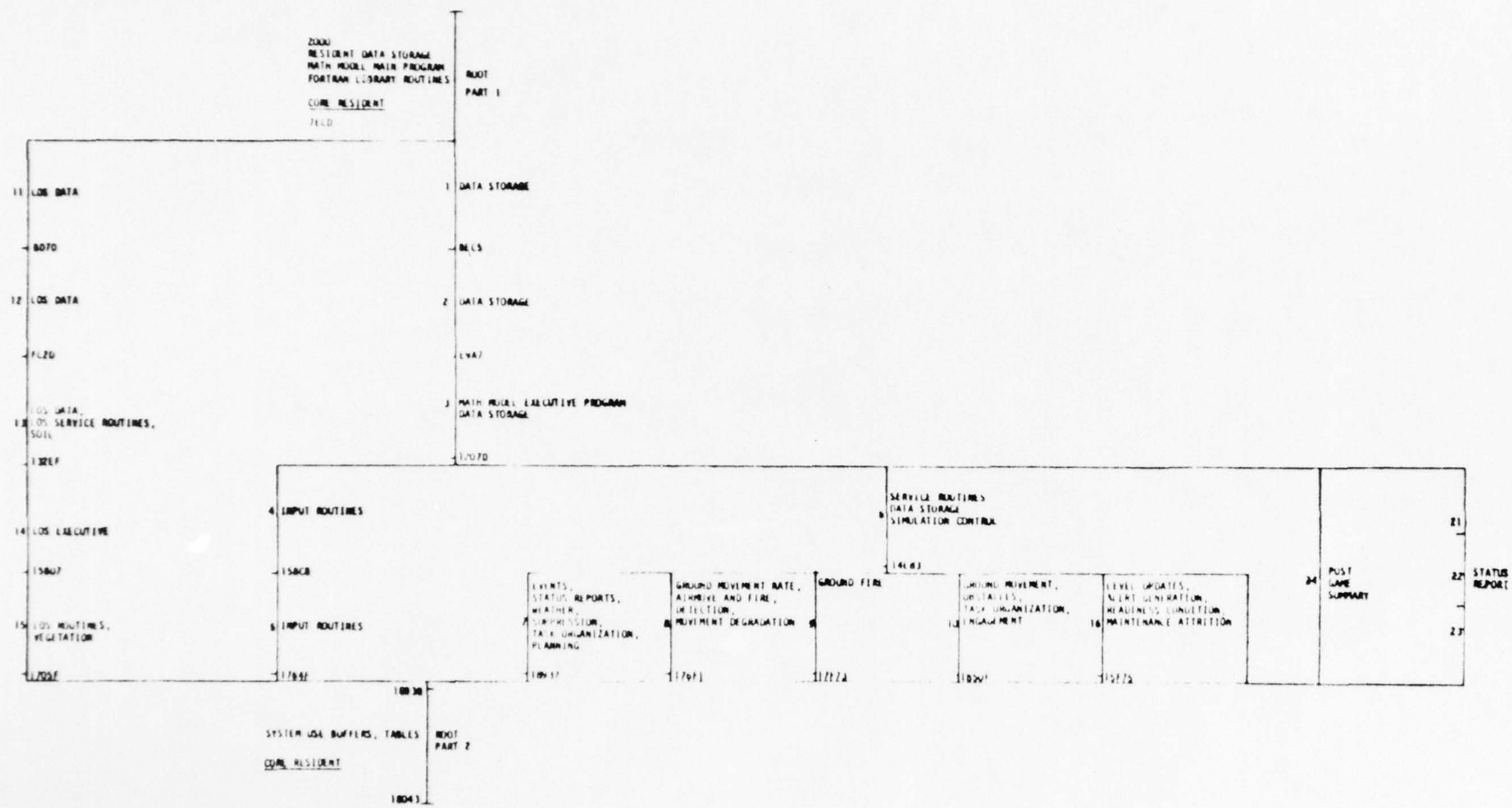


Figure 2-25. Math Model Overlay Structure
(Baseline 16A3-TS-01)

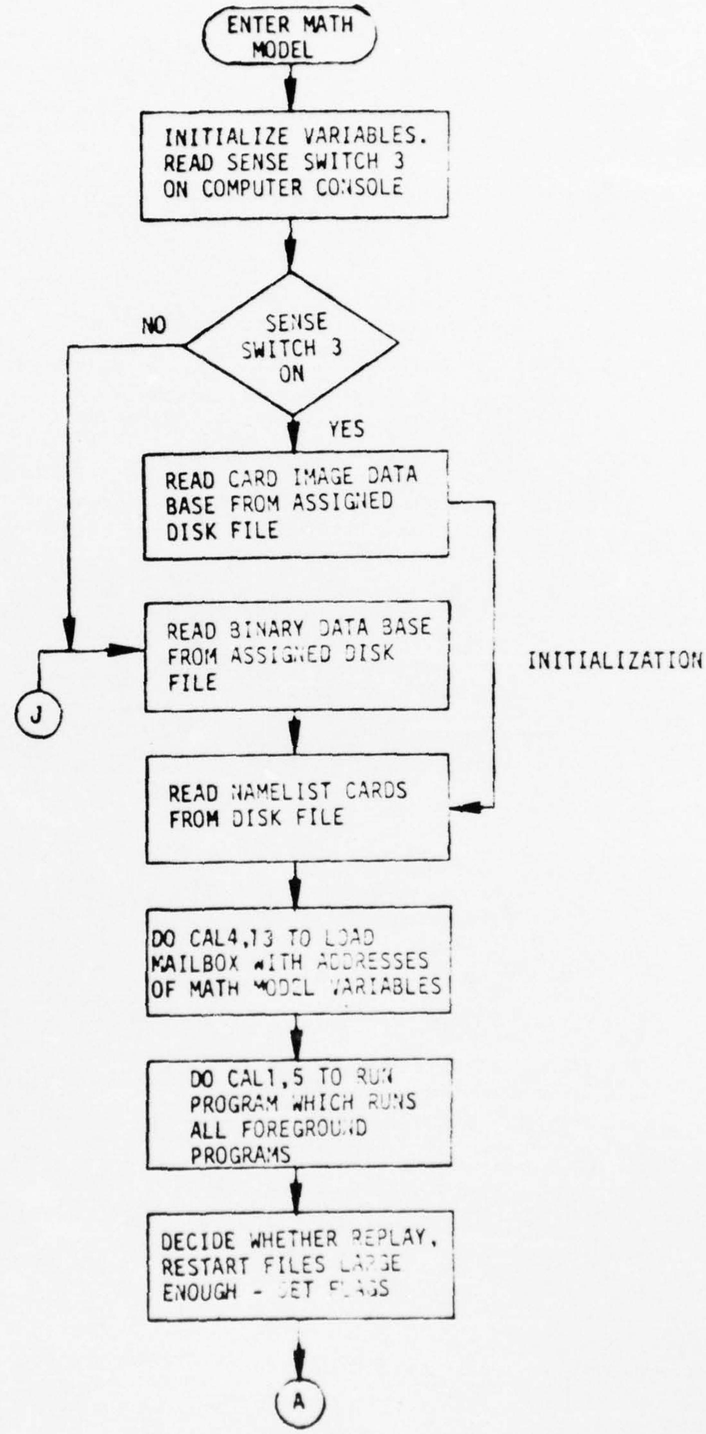
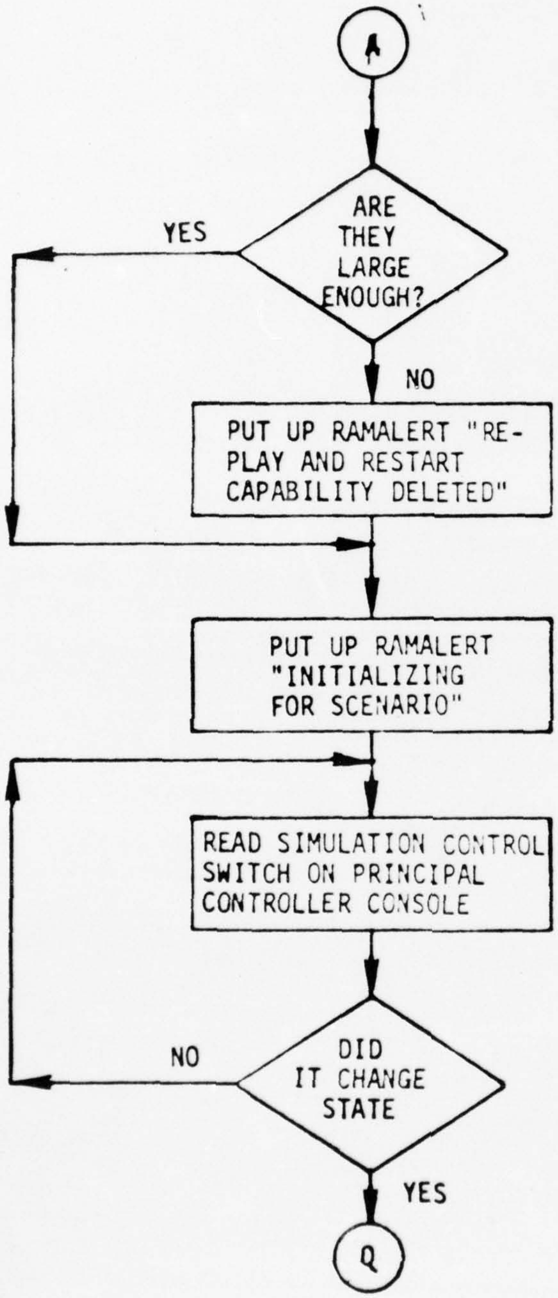


Figure 2-26. Overview of Math Model Operation



DURING THIS TIME ALL CONTROLLERS MAY USE GRAPHIC DISPLAY, MAP-CAMERA, AND COMMAND AND CONTROL SUBSYSTEMS. MATTERS ARE SIMPLIFIED IF ALL DESIRED UNIT DEACTIVATIONS ARE PERFORMED AT THIS TIME.

Figure 2-26. Overview of Math Model Operation, Cont'd.

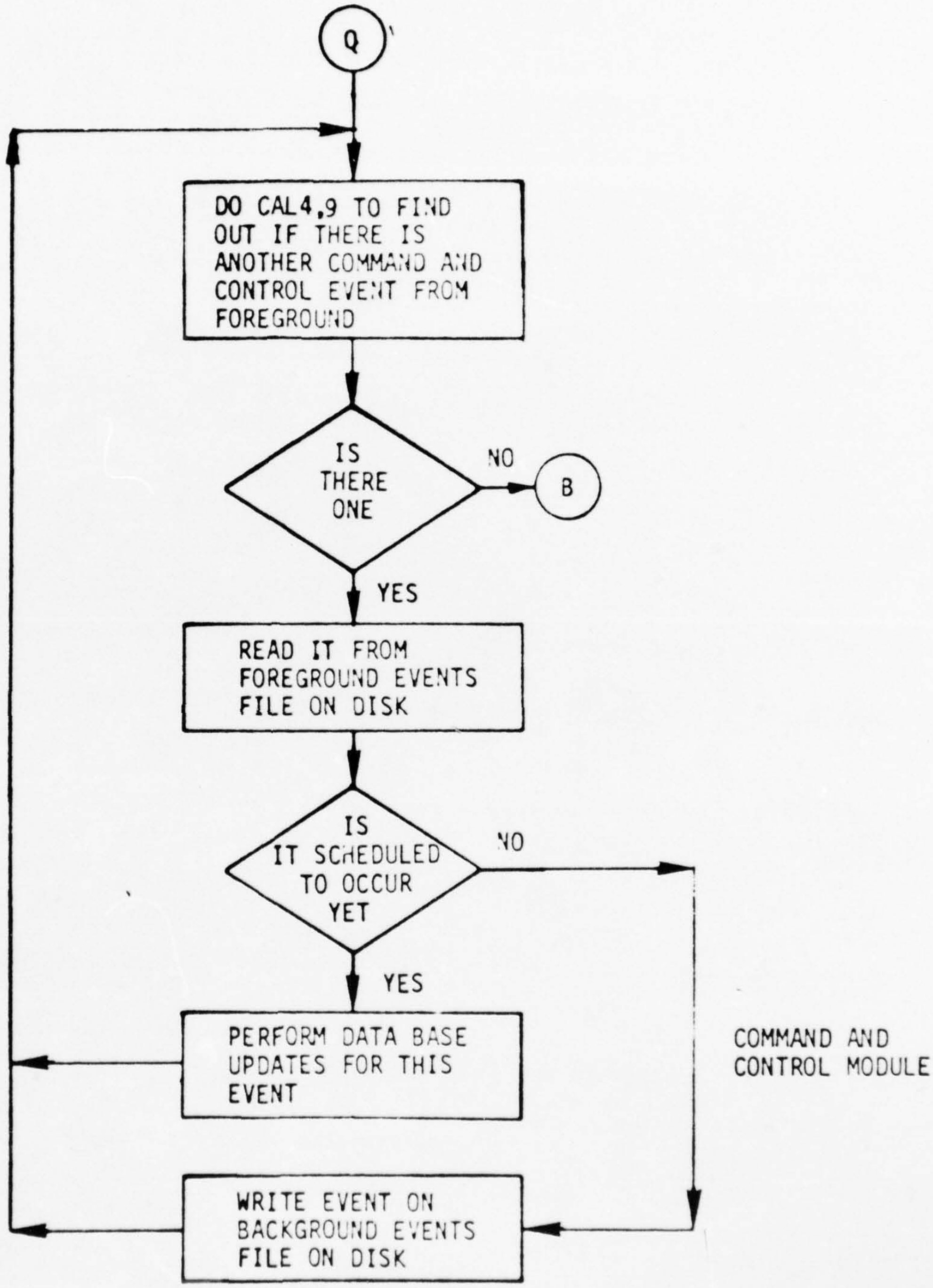


Figure 2-26. Overview of Math Model Operation, Cont'd.

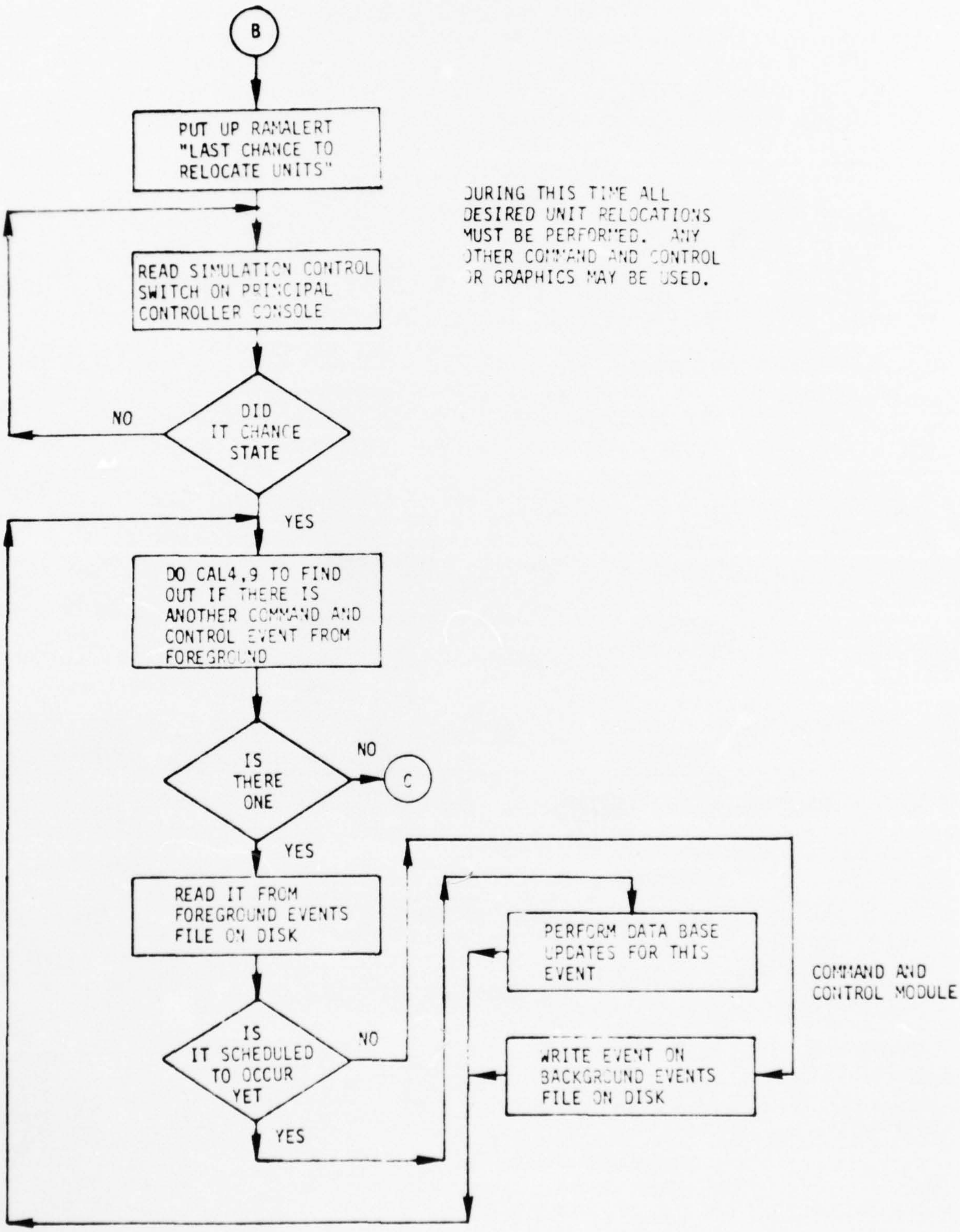


Figure 2-26. Overview of Math Model Operation, Cont'd.

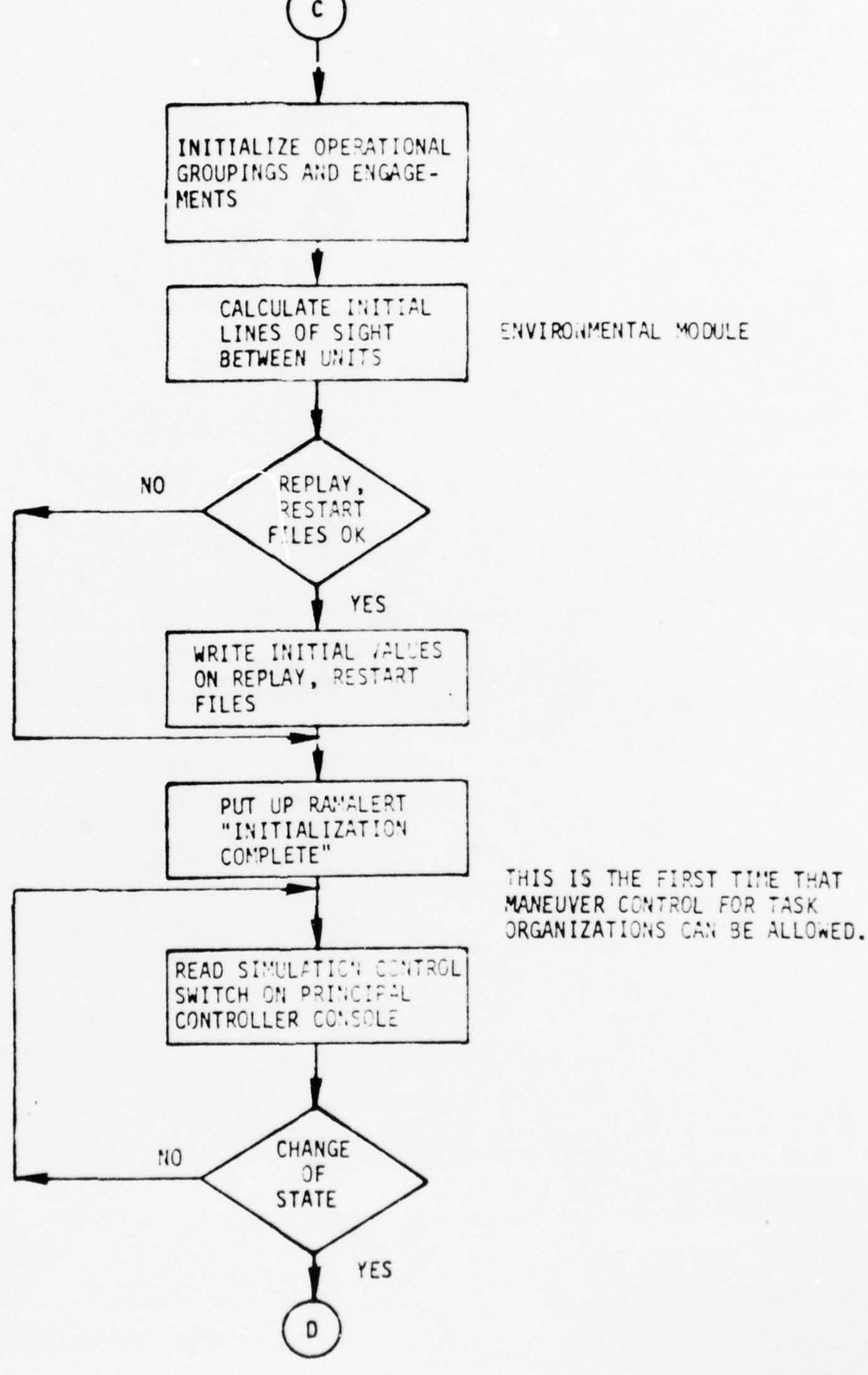


Figure 2-26. Overview of Math Model Operation, Cont'd.

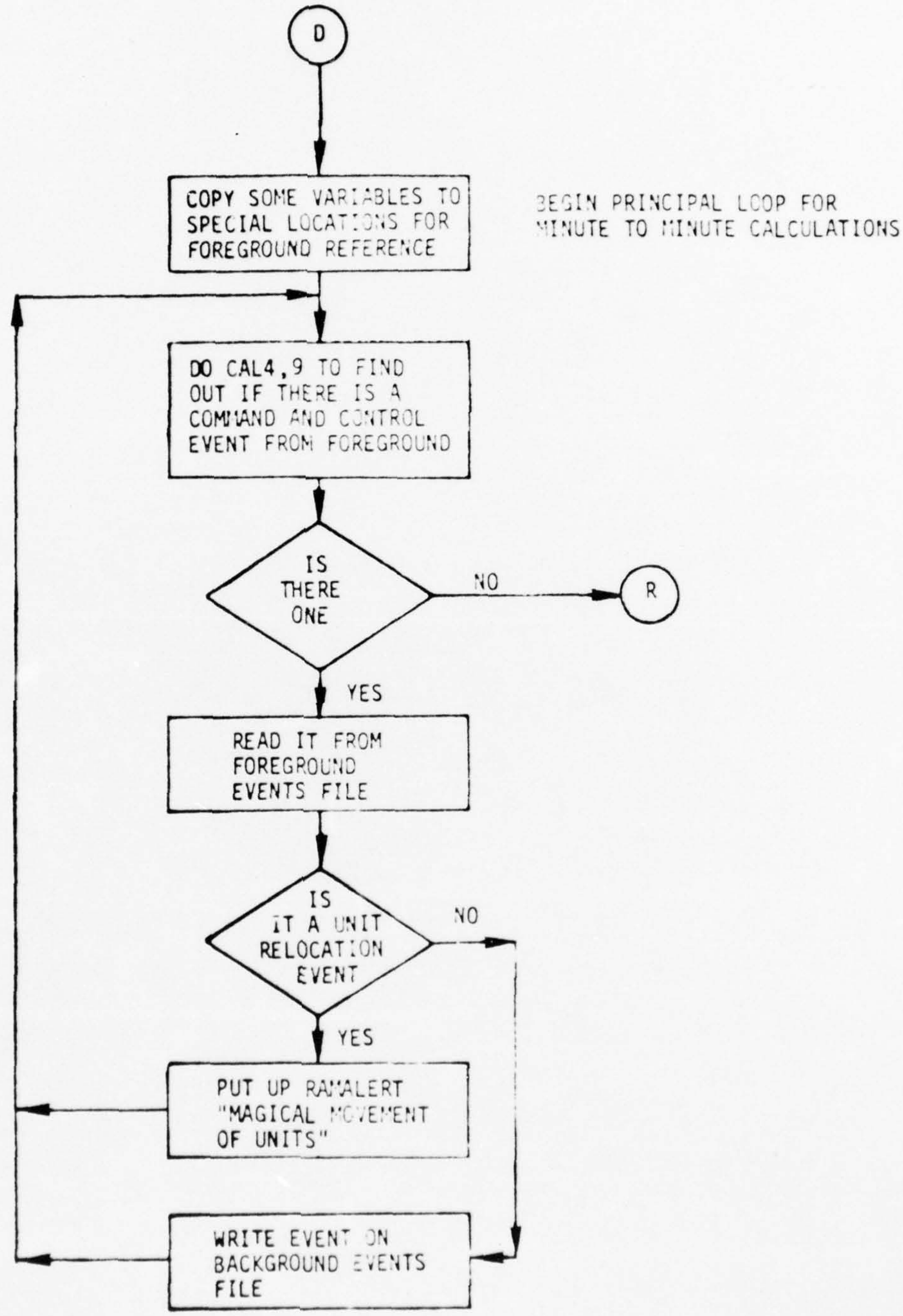


Figure 2-26. Overview of Math Model Operation, Cont'd.

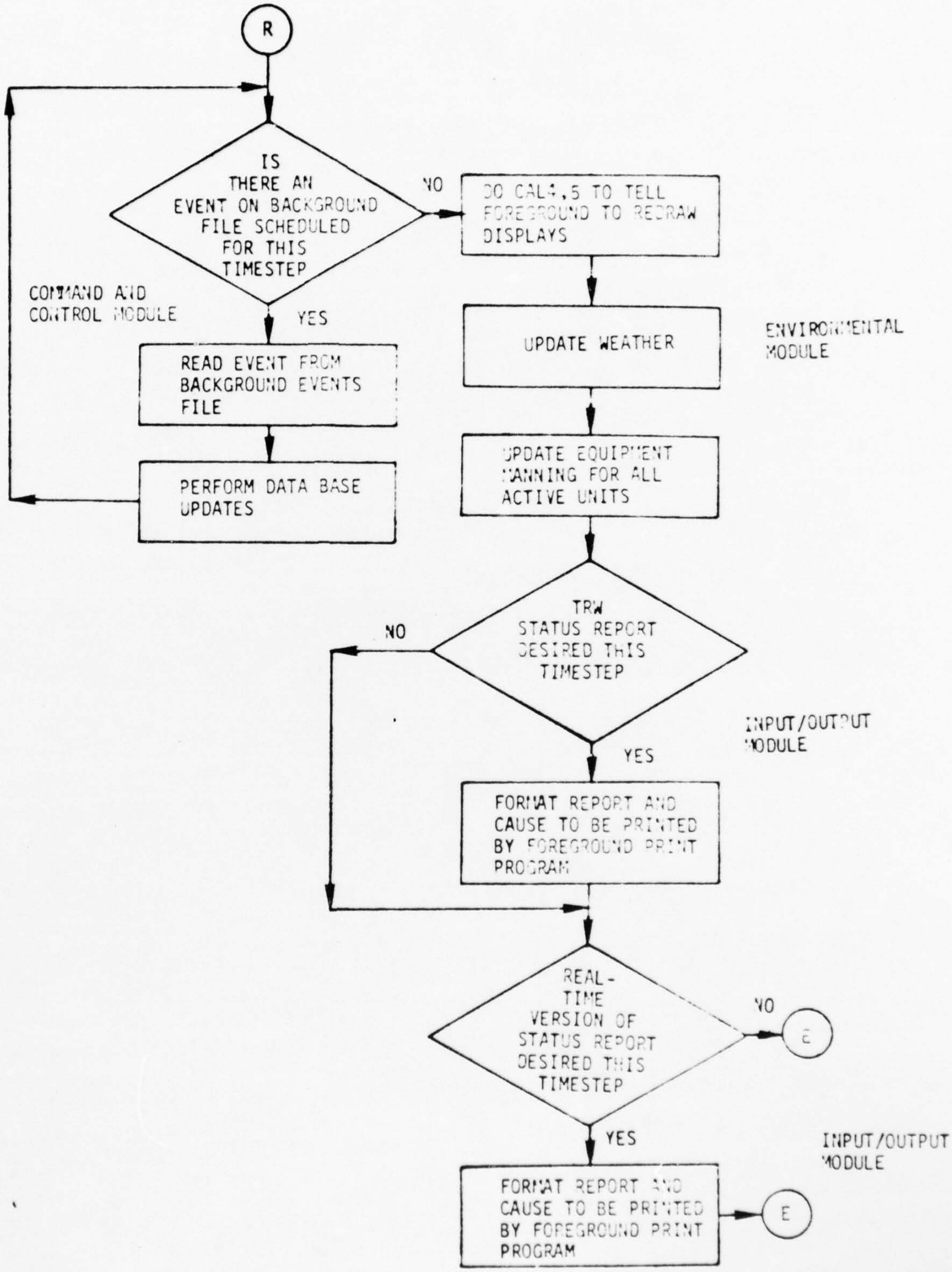


Figure 2-26. Overview of Math Model Operation, Cont'd.

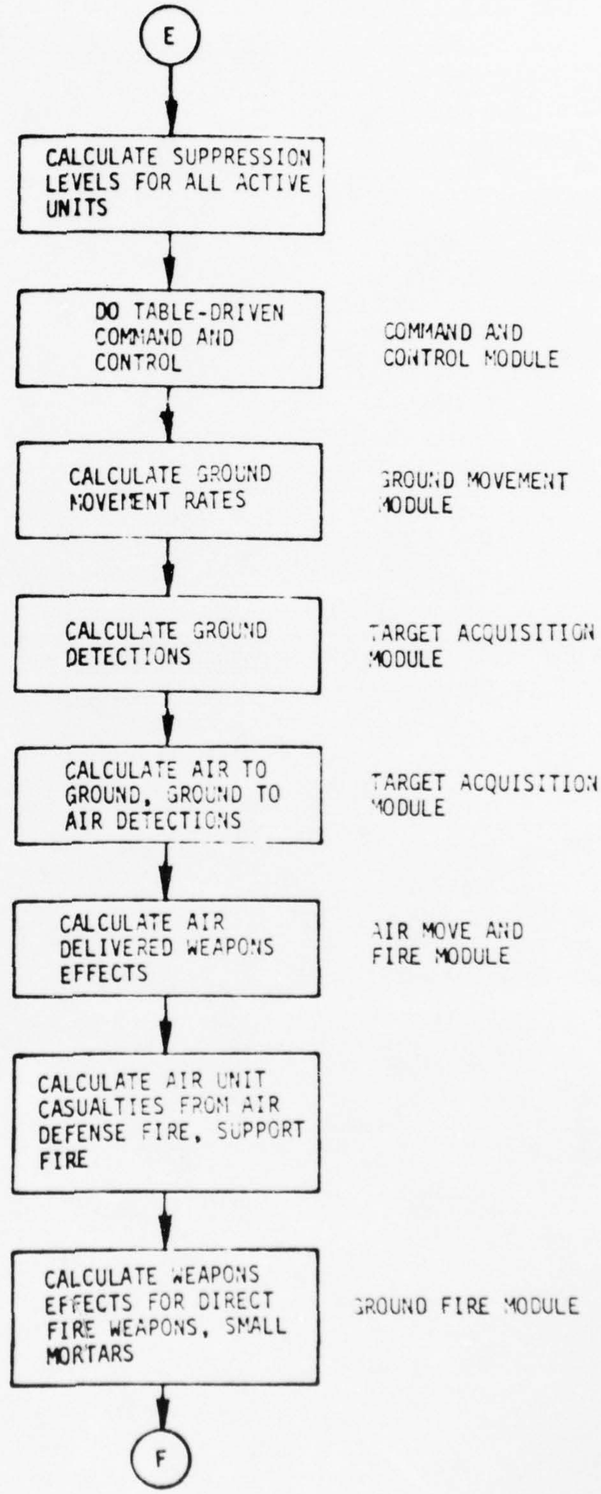


Figure 2-26. Overview of Math Model Operation, Cont'd.

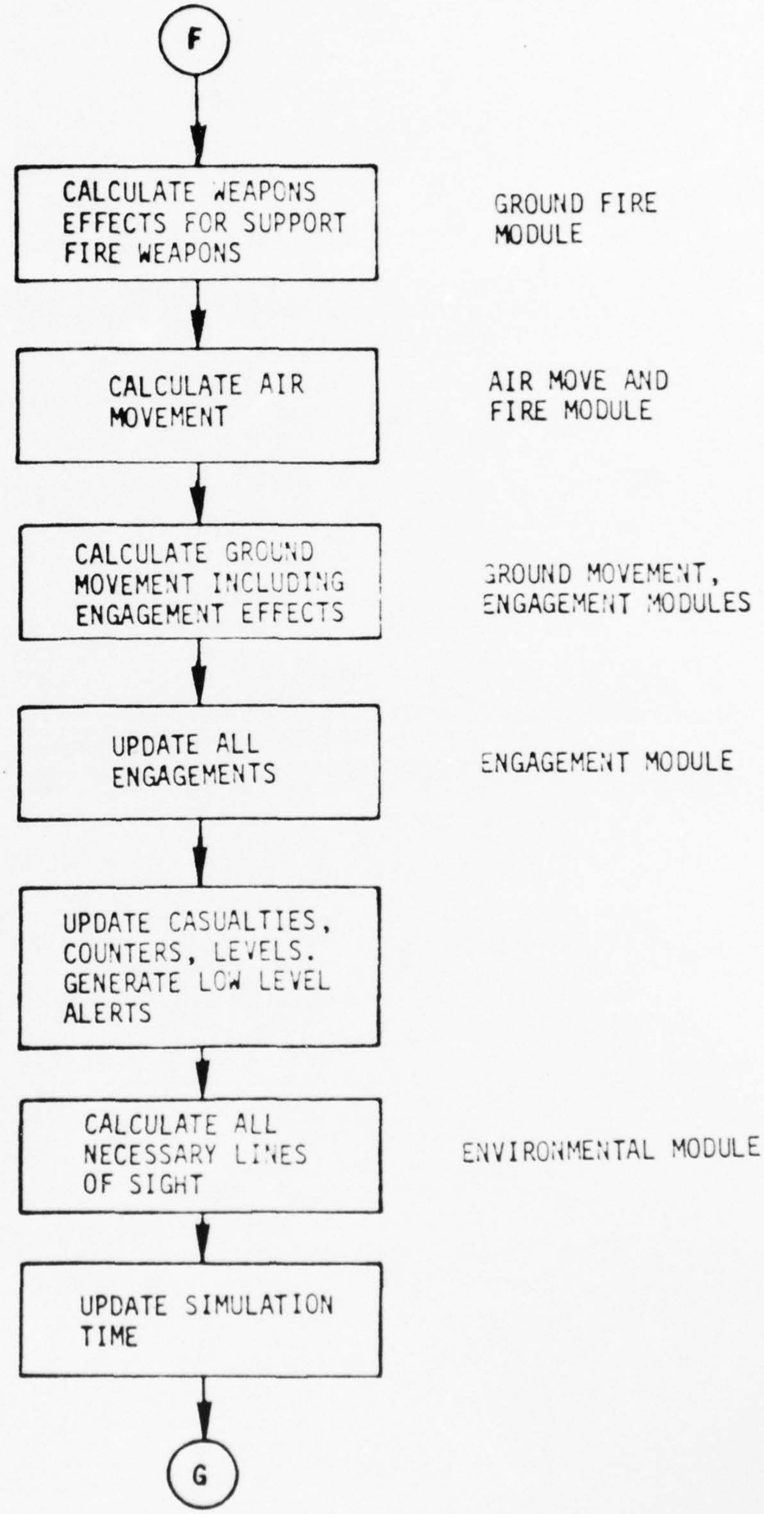


Figure 2-26. Overview of Math Model Operation, Cont'd.

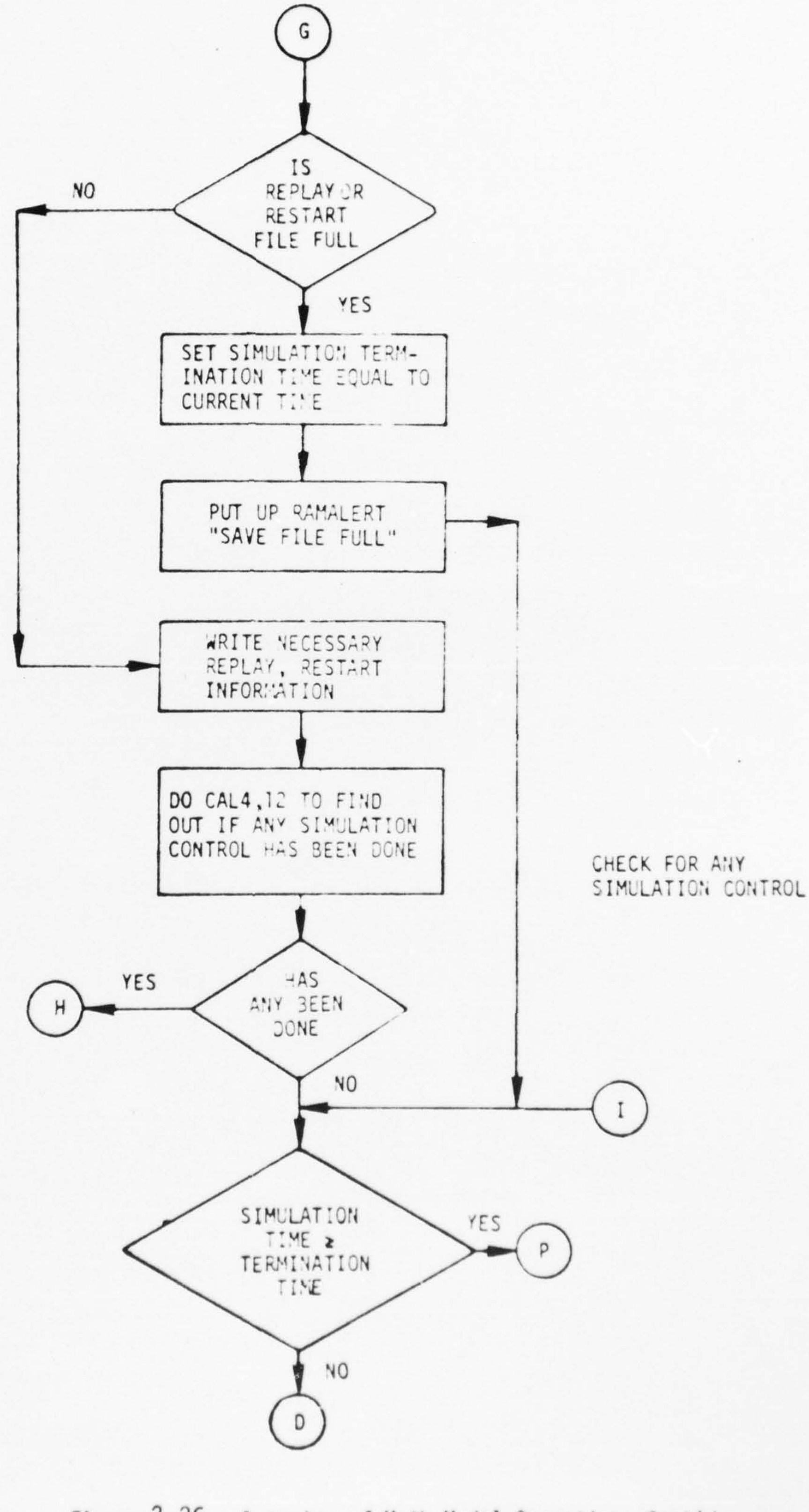


Figure 2-26. Overview of Math Model Operation, Cont'd.

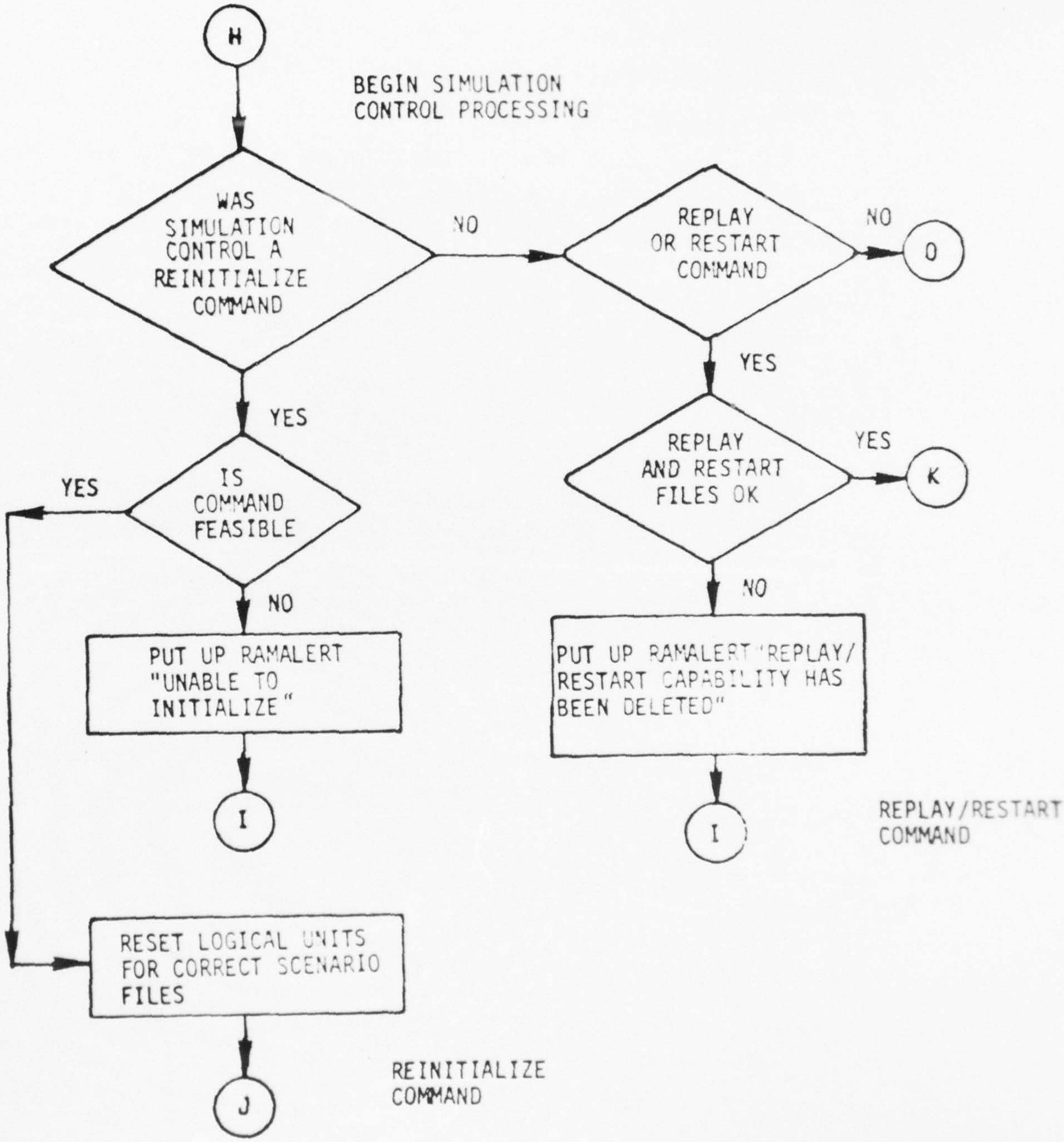


Figure 2-26. Overview of Math Model Operation, Cont'd.

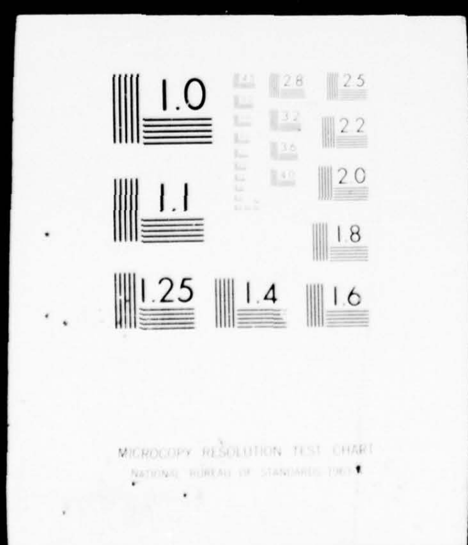
AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

2 OF 8
AD
A038796

2 OF 8

AD
A038796



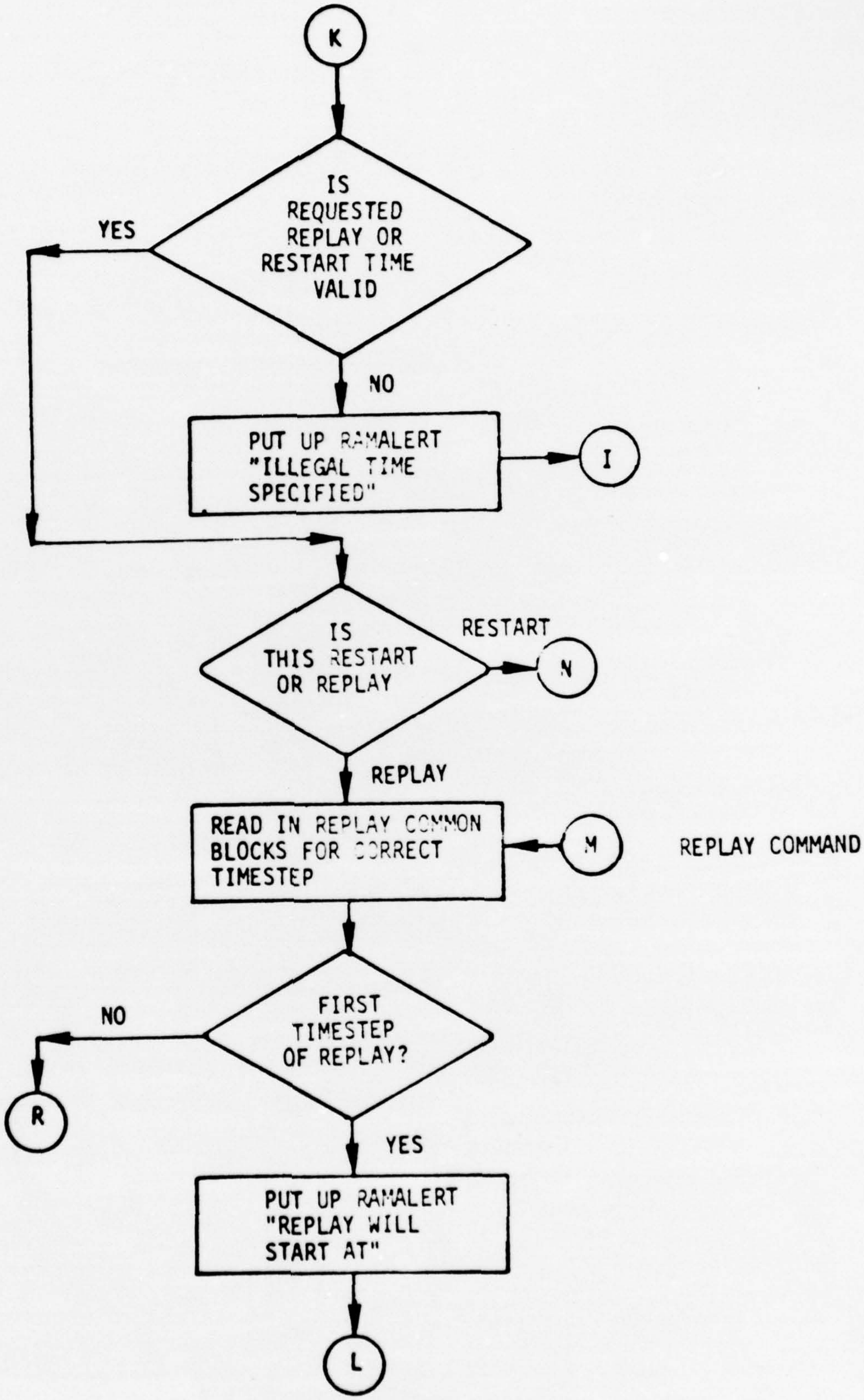


Figure 2-26. Overview of Math Model Operation, Cont'd.

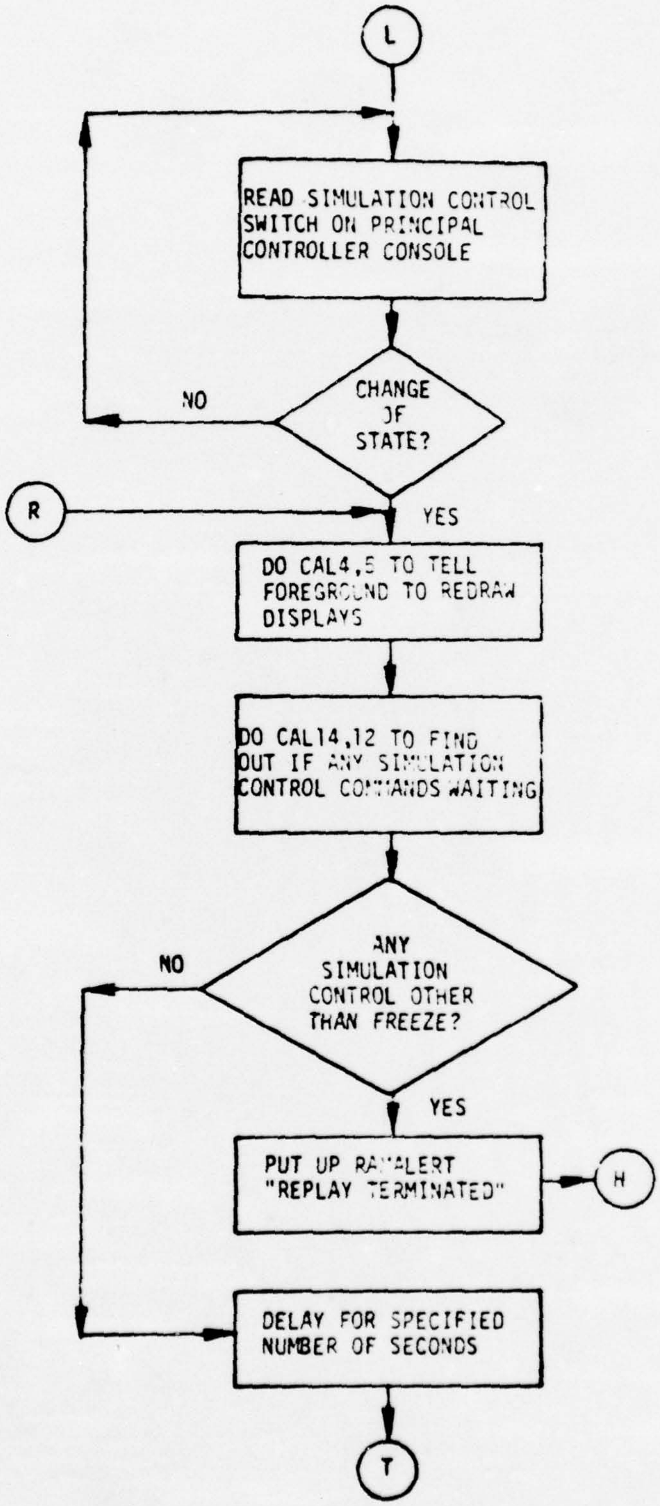


Figure 2-26. Overview of Math Model Operation, Cont'd.

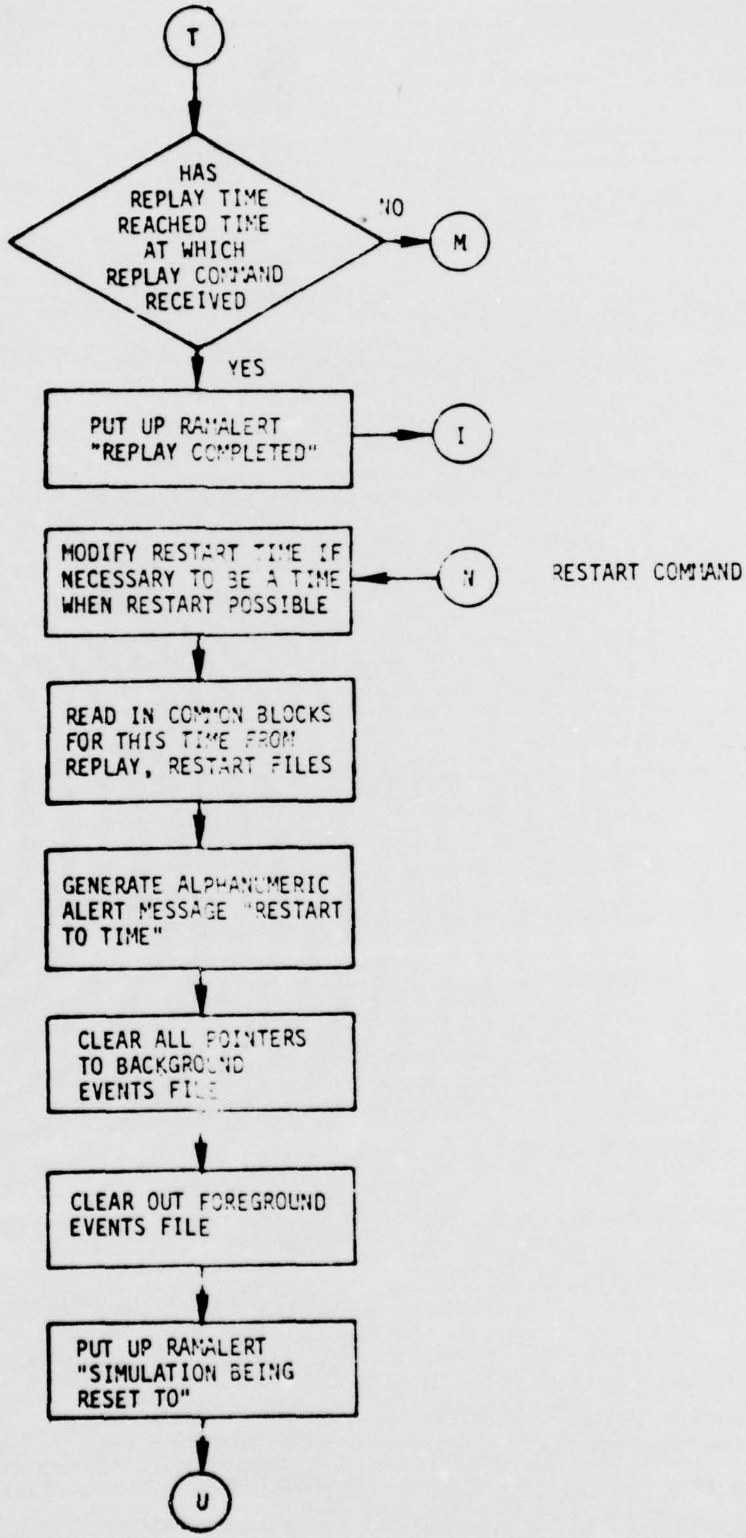
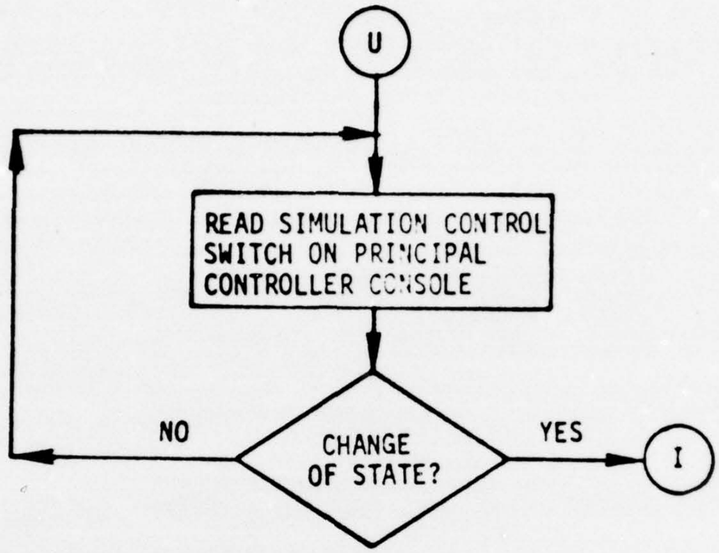


Figure 2-26. Overview of Math Model Operation, Cont'd.



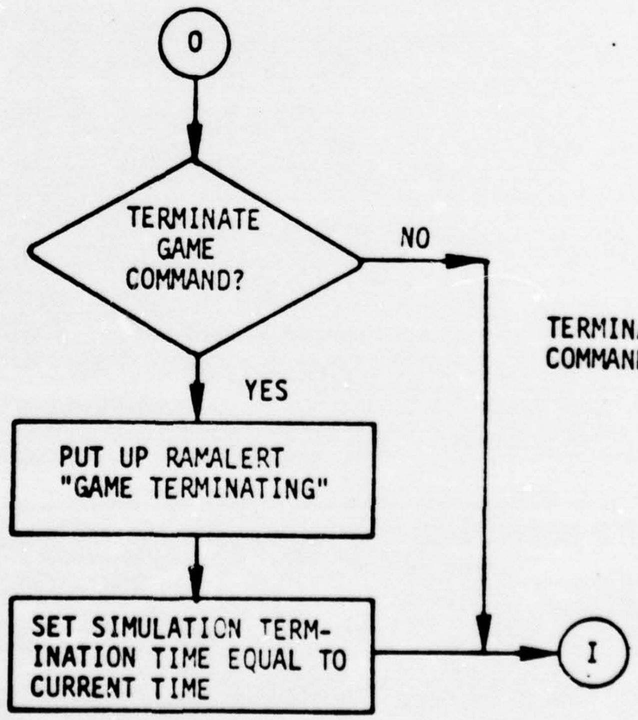
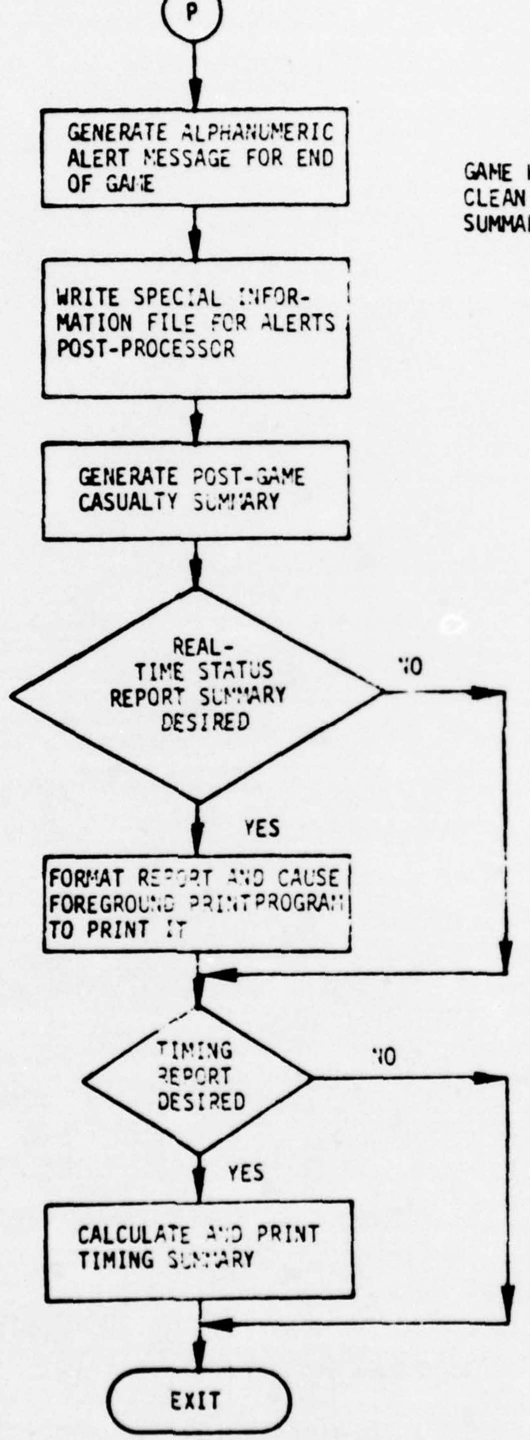


Figure 2-26. Overview of Math Model Operation, Cont'd.



GAME HAS TERMINATED.
CLEAN UP, GENERATE
SUMMARY REPORTS

Figure 2-26. Overview of Math Model Operation, Cont'd.

2.2.3.1.3 Target Acquisition Module

The Target Acquisition Module determines the occurrence of detections between eligible pairs of units and generates alphanumeric alert messages when detections occur. Many environmental and tactical considerations and a wide range of sensor types have been modeled. A detailed description of this module is presented in Section 5.3.

2.2.3.1.4 Ground Fire Module

The Ground Fire Module is a complex, detailed model which allocates and controls the fire of all ground weapons modeled in CATTS. It computes firing rates, casualties, and ammunition expenditures for each weapon in each unit each time step. A detailed description is provided in Section 5.4.

2.2.3.1.5 Ground Movement Module

The Ground Movement Module controls and directs the movement of all ground units in the area of operations. At each mathematical model timestep, each unit is examined to determine whether it should start or stop moving. For moving units, a movement rate is calculated based on tactical considerations, existing and new engagements, suppression, and environmental factors. Units may move singly or as part of an operational grouping. The disruptive effects of obstacles to movement are also modeled. A more detailed description is provided in Section 5.5.

2.2.3.1.6 Engagements Module

The purpose of the Engagements Module is to cause ground units in the model to respond in a tactically realistic way to enemy fire and/or proximity. It determines when units will fire direct fire weapons, when they will form engagements, and when they will break them off. A more detailed description is provided in Section 5.6.

2.2.3.1.7 Input/Output Module

The Input/Output Module consists of the portions of the model which are concerned with input or output. It includes those routines which initialize the model data base, those which produce the line printer status report,

those which generate alert messages, and those which produce Ramalerts. A detailed description is provided in Section 5.7.

2.2.3.1.8 Air Module

The Air Module updates location, direction, speed, and altitude of each air unit according to individual, input flight plans. This occurs at intervals of one quarter-minute or less. For each quarter-minute air/ground interactions are calculated, including detections, firing, air weapons delivery, and casualty assessment. A more detailed description is provided in Section 5.8.

2.2.3.1.9 Command and Control Module

The Command and Control Module performs the necessary data base updates for both the interactive and the table-driven command and control in the math model. The interactive portion processes the command and control event notices received from the foreground Command and Control Program. The table-driven portion uses a tabular set of input decision rules which determine changes in unit status if the conditions specified in the table are met. A more detailed description is provided in Section 5.9.

2.2.3.1.10 Miscellaneous and Ancillary Module

The Miscellaneous and Ancillary Module is a set of service routines used by other modules to perform common calculations of such things as line intersections. Section 5.10 contains a description of each.

2.2.3.2 Foreground Software

The complexity of the controller interactions required in CATTS dictated that large, complex software packages would be required. To increase modularity and speed development, each major interactive function was designed as a separate program. Because the interactive software needs to be interrupt driven, they run as foreground programs, and are often referred to as "foreground software". An overview of each of the foreground programs and data areas will be presented as the remainder of this section. Detailed descriptions of each may be found in Section 4 of the Programming Report.

2.2.3.2.1 CAL4 Handler

The CAL4 handler is actually a group if small programs which accomplish the user written CAL's. (A CAL is System Function Call instruction which is used to transfer control to a Xerox RBM monitor task. RBM does allow the user to write some tasks which are treated as a system call but are processed by user-written software.) See the Xerox RBM Reference manual for a general explanation. When a CAL4 instruction is executed, the monitor traps to location X'4B' which contains an XPSD instruction to the CAL4 handler. The CAL4 handler decides which CAL4 this is, branches to the correct routine for this CAL, processes the CAL, and returns to the calling routine. Since the CATTS CAL4's are essentially non-interruptible, the processing performed by one must necessarily be minimal - in the case of CATTS they are used primarily to communicate small pieces of information between two programs. As an example, the mathematical model uses a CAL4 to set a flag in the mailbox which tells Graphic Display that a model timestep has occurred and that all displays should be recalculated. Table 2-1 shows all the CAL4's and their functions.

2.2.3.2.2 Mailbox

The Mailbox is a common data storage area for all the various programs. The addresses of the mathematical model variables needed by the foreground programs are stored here. Also stored here are a number of flags used for communications between programs - for example, Command and Control set a flag in the mailbox to inhibit Graphic Display while a menu is being used, to avoid having the menu display erased or overwritten. A layout of the mailbox is shown in Section 3.75.

2.2.3.2.3 Graphic Display Program

The Graphic Display program contains the software necessary to create color video graphic displays on the television monitors at each console. These displays are designed to look like normal military map overlays. The selection of "overlay", or display, types is controlled by a set of alternate action switches at each console. Since the map displays on each monitor are controlled by the pan, tilt, and zoom of the camera corresponding to that

Table 2-1. Use and Purpose of CAL4's*

CAL4	PURPOSE	LOCATION
0	Connects other CAL4's	CAL4NEW
1	Synchronous print handling	PRINTQ
2	Ramtek handler	RAMT
3	Request for audio tape time	CANDC(Command&Control)
4	Ramalert handler	GANDC(Command&Control)
5	Time step indicator	GDEXEC(Graphic
6	Alert request	Alerts
7	Camera location data	MAP/VIDEO
8	Ramtek transfer data	RAMT(used by CANNED)
9	Request C&C	CANDC(Command&Control
10	Write direct	CAL4NEW
11	Trigger C&C executive	CANDC(Command&Control
12	Transfer simulation control data	CANDC(Command&Control)
13	Initialize pointers to math model variables	CAL4NEW
14	Timing routine	CANDC(Command&Control)
15	Read direct	CAL4NEW

*RBM (the Xerox operating system used in CATTS), allows the user to connect various load modules by use of what is referred to as CALS. In CATTS, a CAL2 is used to request that a status report be written to disk. CAL4's allow communication between load modules and interrupt levels (tasks).

console, the graphics displayed have to be calibrated to the current camera position. Table 2-2 lists the display types available to each controller. Figure 2-27 shows the communication between the graphic display software and other system elements. Figure 2-28 presents a simplified flow diagram of the graphic display program. The overlay structure of the graphic display program is shown in Figure 2-29.

2.2.3.2.4 Command and Control Program

The Command and Control software provides the interactive command and control simulation control capabilities of the CATTS system. Tables 2-3 and 2-4 provide summaries of those capabilities. It also produces the Ramalert messages generated by the mathematical model. These alerts are treated as if they were command and control menus.

The Command and Control software is a critical part of the CATTS system since it provides the dynamic tactical response needed for a meaningful exercise and greatly facilitates system operation and performance evaluation for the controllers.

Figure 2-30 shows an overview of communications between the command and control program and other system elements. As the figure shows, this is perhaps the most complex software interface of the CATTS system. Response-time requirements for the cursor showing graf-pen location led to a design in which there are actually two programs which run at different interrupt priority levels.

The Command and Control polling routine is triggered by a clock interrupt at intervals of 1/20 second. It runs at a high interrupt level and consequently must do minimal processing. What it does do, as shown in Figure 2-31, is:

- For each active graf-pen
    - -- spark pen every other entry
    - -- read pen location and display cursor every non-spark entry
    - -- read tip-switch to detect menu selection. If selection made, then trigger C&C executive.

Table 2-2. Types of Color Graphic Displays Provided by CATTS

Map Grid Coordinates
Tactical Overview
Red/Blue Minefields
Red/Blue Obstacles
Red/Blue Front-Line Trace
Area Occupied by Red/Blue Combat Units
Area Occupied by Red/Blue Combat Support Units
Area Occupied by Red/Blue Combat Service Support Units
Command Posts of Red/Blue Combat Units
Command Posts of Red/Blue Combat Support Units
Command Posts of Red/Blue Combat Service Support Units
Direction of Movement of Red/Blue Combat Units
Direction of Movement of Red/Blue Combat Support Units
Direction of Movement of Red/Blue Combat Service Support Units
Red/Blue Ground Radar Devices
Red/Blue Ground Sensors
Red/Blue Observation Posts
Red/Blue Night Vision Devices
Red/Blue Airborne Sensors
Coverage of Red/Blue Ground Radar Devices
Coverage of Red/Blue Ground Sensors

Coverage of Red/Blue Night Vision Devices

Red/Blue Antitank Rockets
Red/Blue Antitank Missiles
Red/Blue Artillery Weapons
Red/Blue Air Defense Weapons
Red/Blue Mortars
Red/Blue Air Strikes
Red/Blue Preplanned Targets
Red/Blue Impacting Fires
Red/Blue Platoon Control Measures
Red/Blue Company Team Control Measures
Red/Blue Battalion Task Force Control Measures
Red/Blue Brigade Control Measures
Red/Blue Division Control Measures

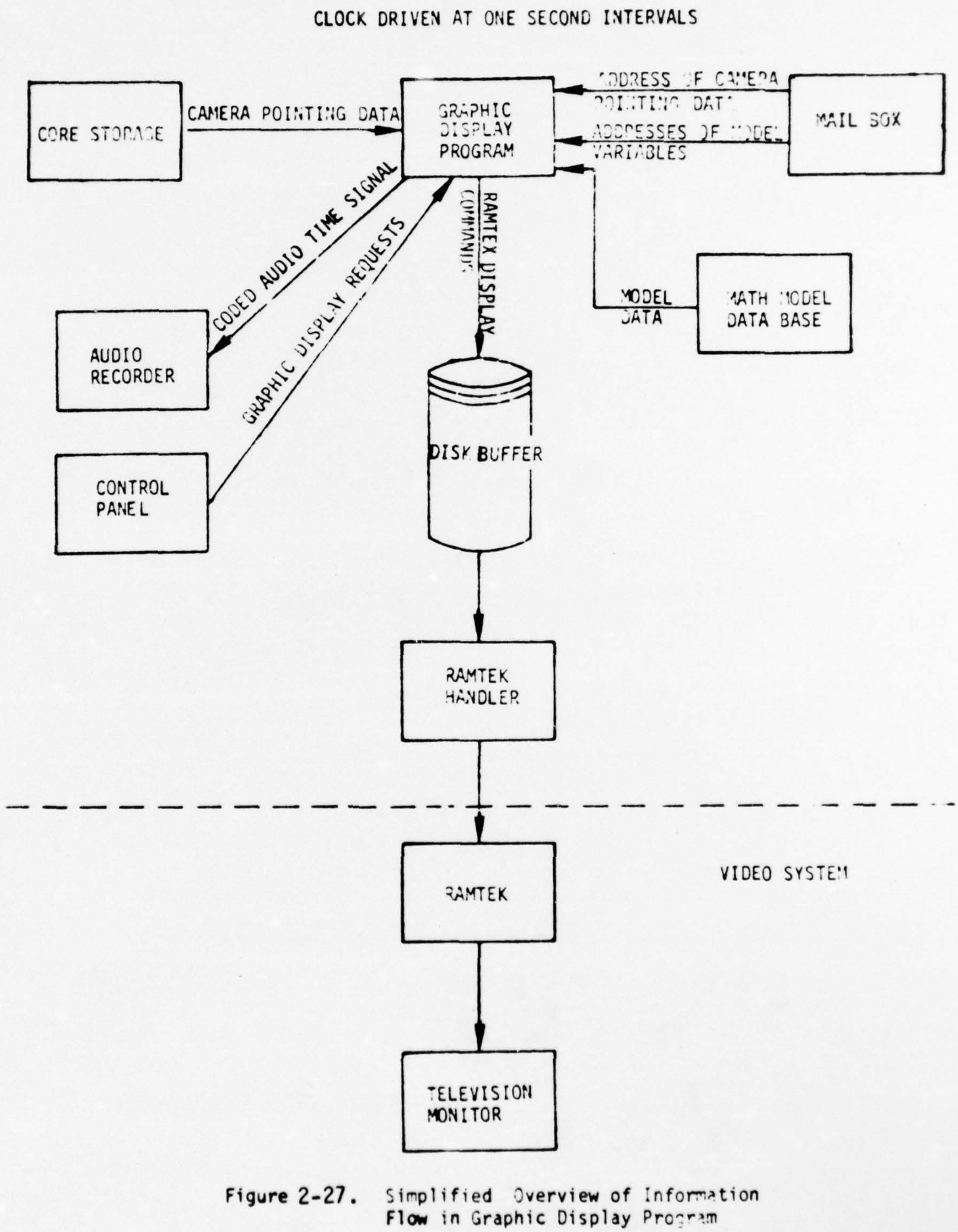


Figure 2-27. Simplified Overview of Information Flow in Graphic Display Program

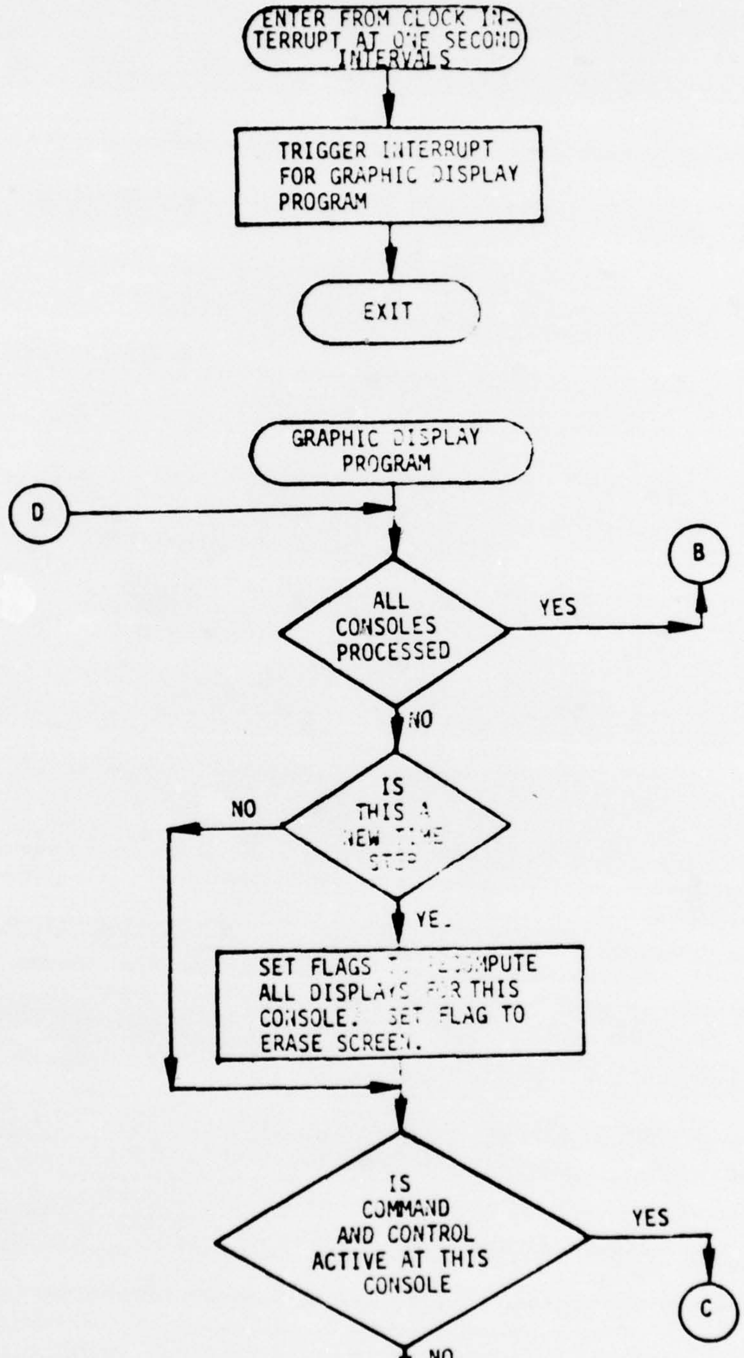


Figure 2-28. Overview of Graphic Display Program

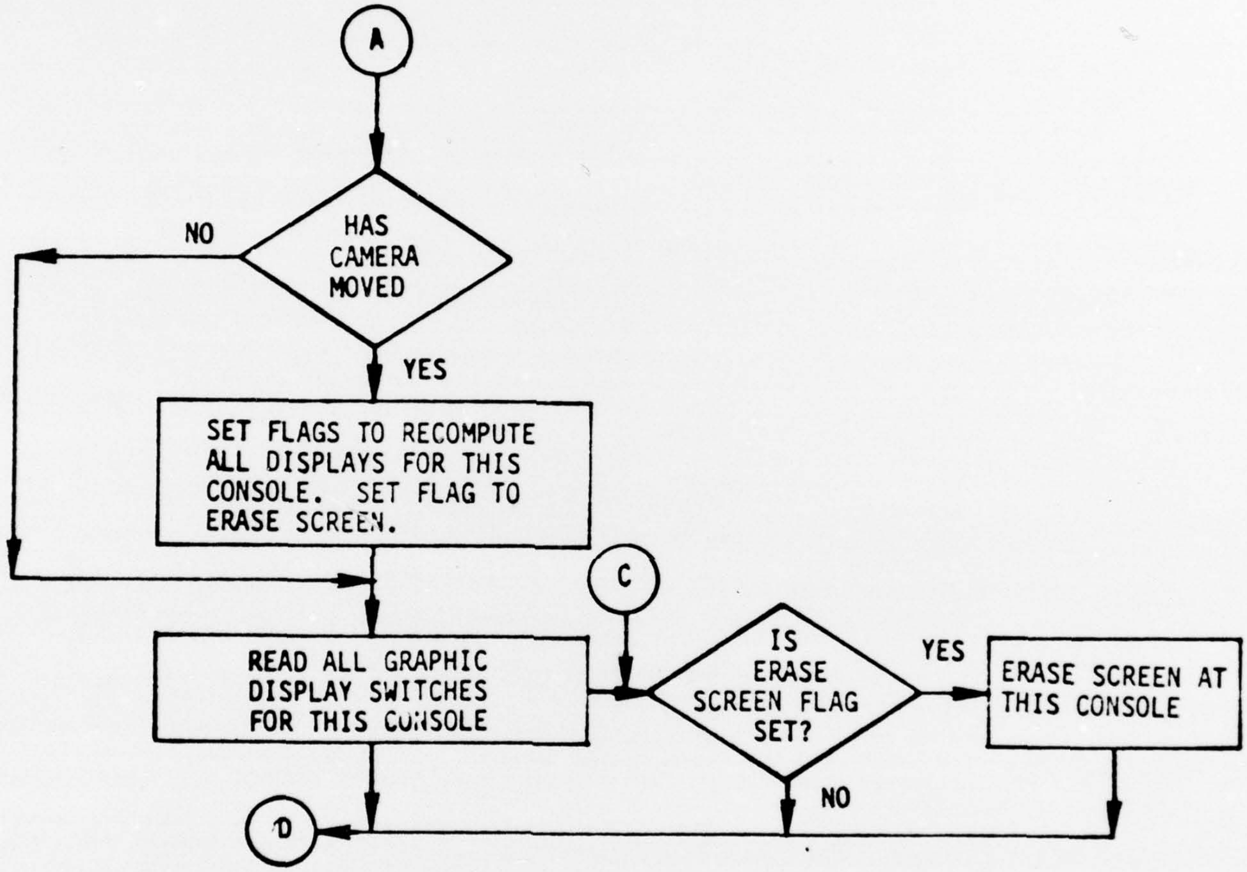


Figure 2-28. Overview of Graphic Display Program, Cont'd.

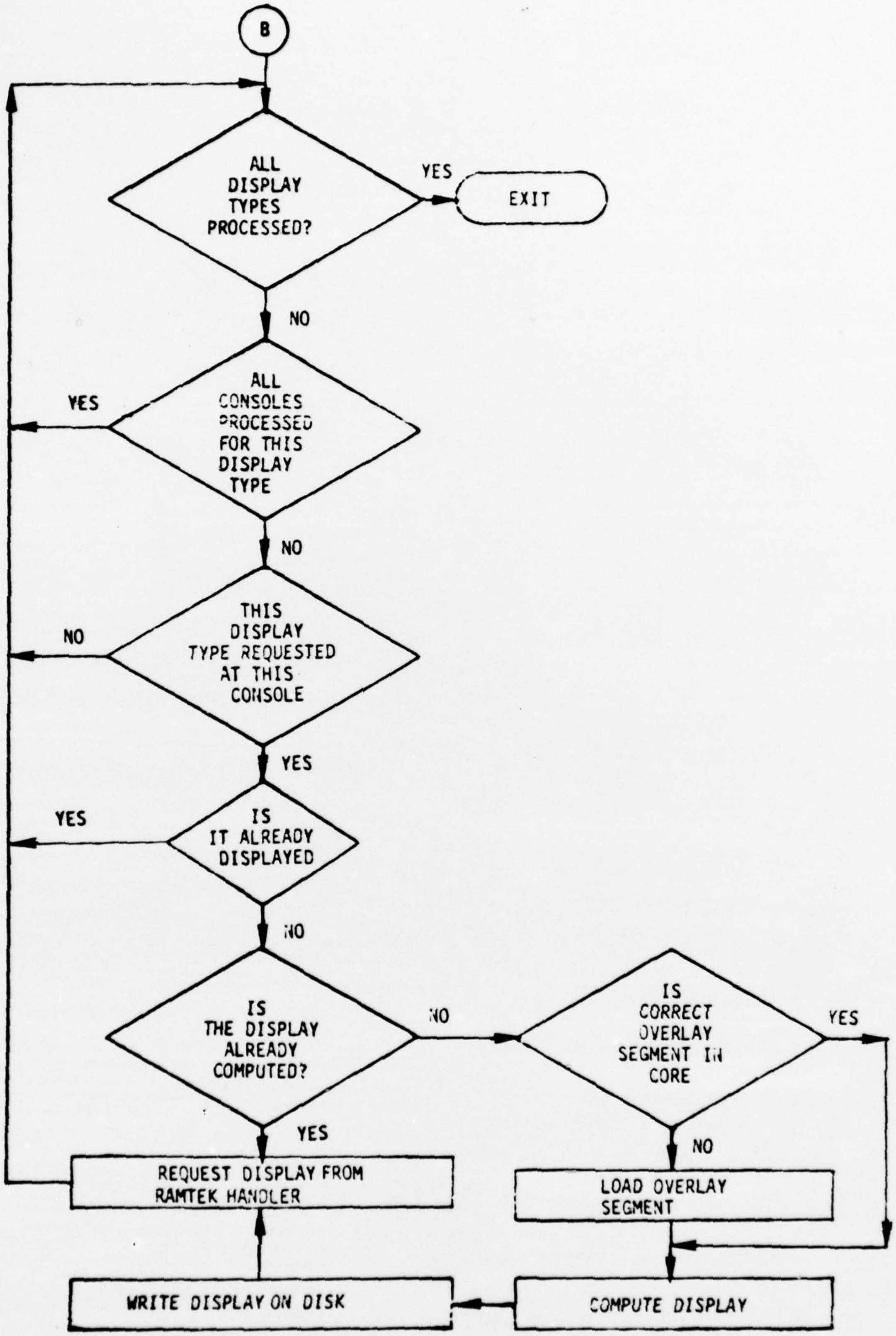


Figure 2-28. Overview of Graphic Display Program, Cont'd.

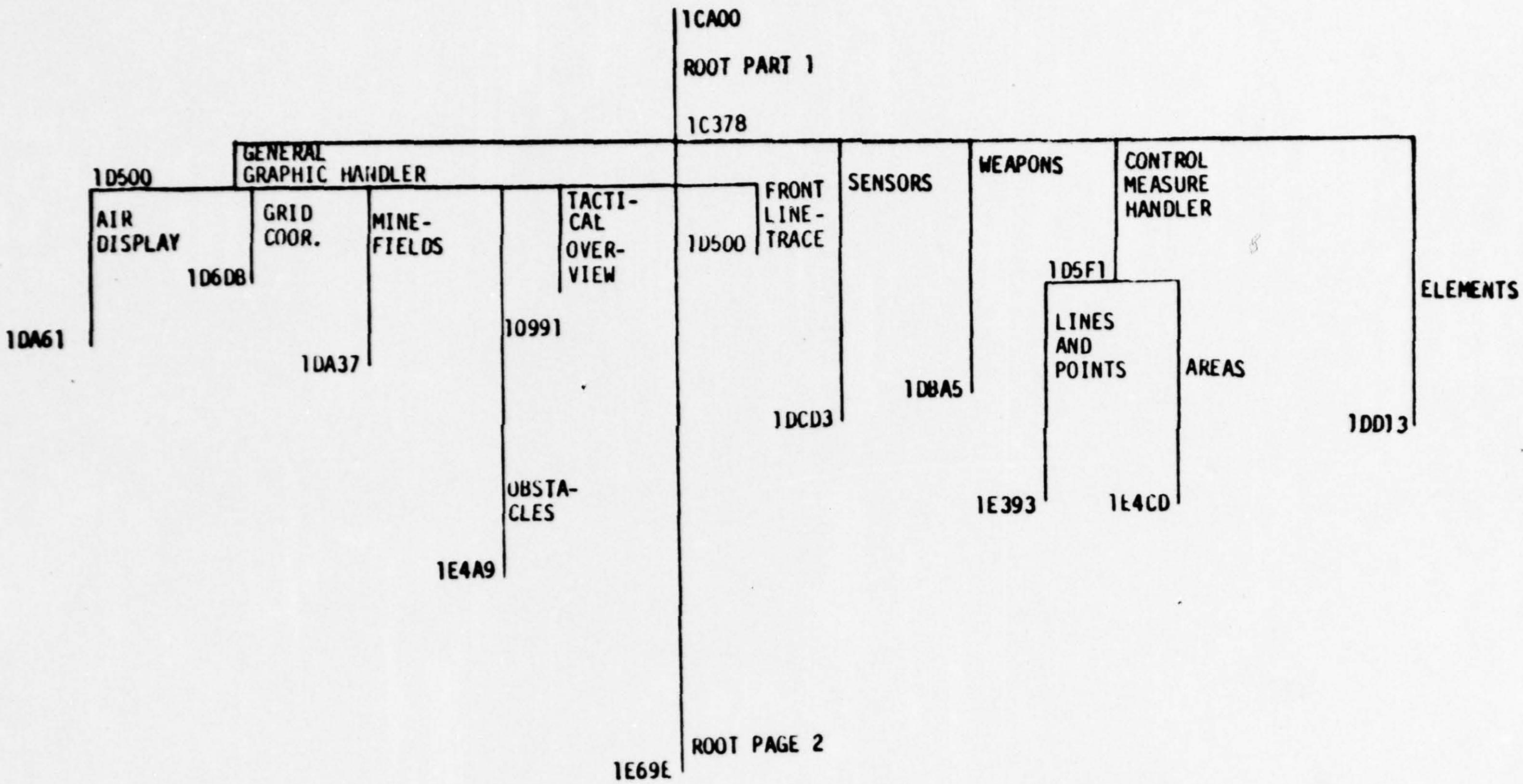


Figure 2-29. Graphic Display Overlay Structure (Baseline 16A3-TS-01)

Table 2-3. Command and Control Capabilities Provided by the CATTS System

- Change the global weather class
- Execute a preplanned mission
- Deactivate red or blue units
- Resupply red or blue units
- Perform red or blue task organization
- Create a red or blue air strike or air recon mission
- Issue red or blue air defense commands
- Maneuver red or blue units or task organizations
- Create, move, delete red or blue control points, lines, or areas
- Relocate red or blue units instantly
- Issue red or blue fire/no fire commands for any or all weapons in a unit

Table 2-4. Simulation Control Capabilities Provided by the CATTS System

- Reinitialize the game to any pre-defined scenario.
- Back up to an earlier point in the current game and restart from there.
- Back up to an earlier point in this game and replay it exactly as it happened. Allow full grapnics interactive capability. Replay at controller-specified speed (if no audio replay desired).
- Terminate the replay currently in progress.
- Terminate this game. Print pre-specified post-game summaries.
- Freeze this game until further notice. Allow full interactive graphics and command and control capabilities during freeze, but do not allow model to execute.

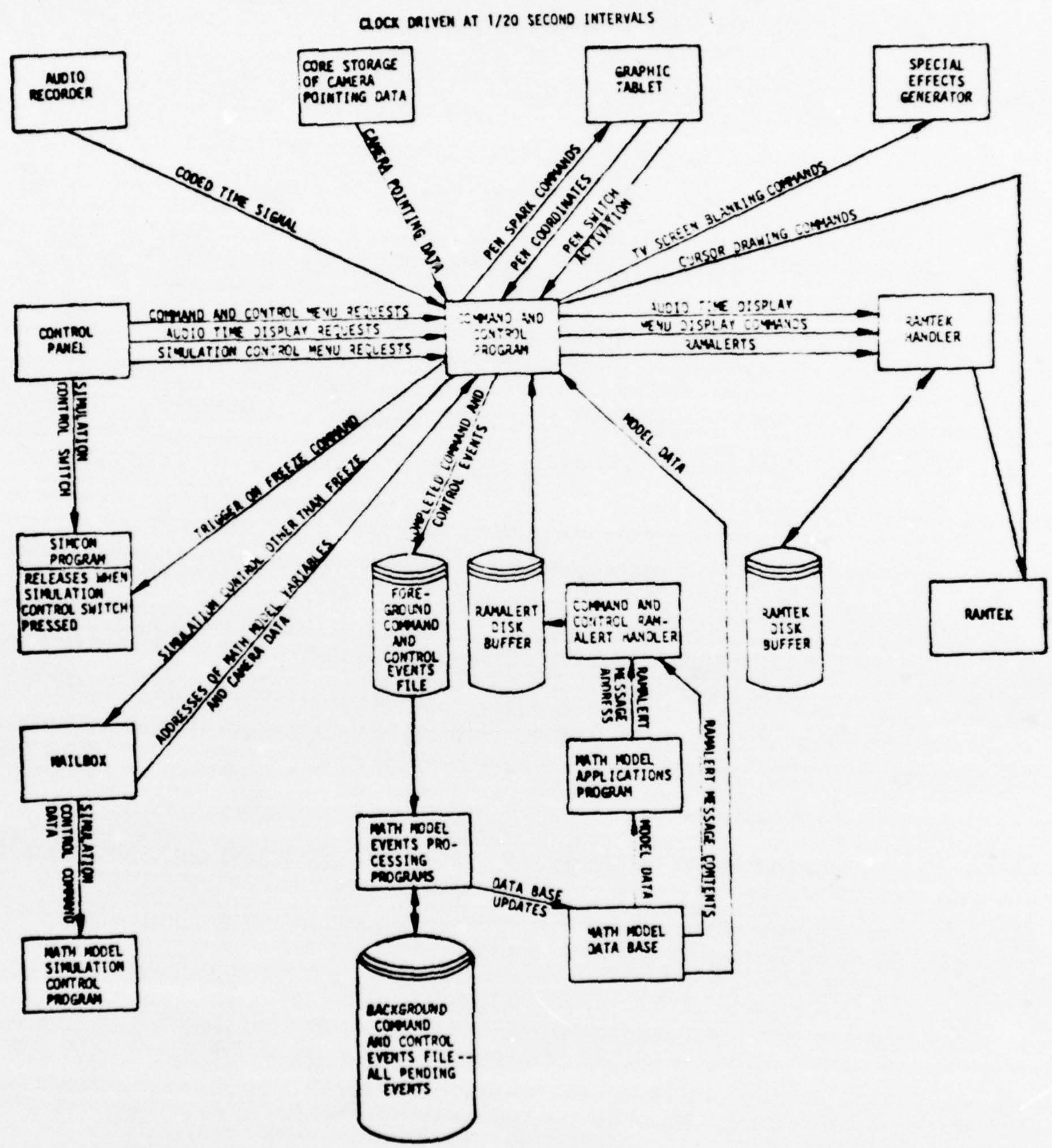


Figure 2-30. Simplified Overview of Information Flow in Command and Control Programs

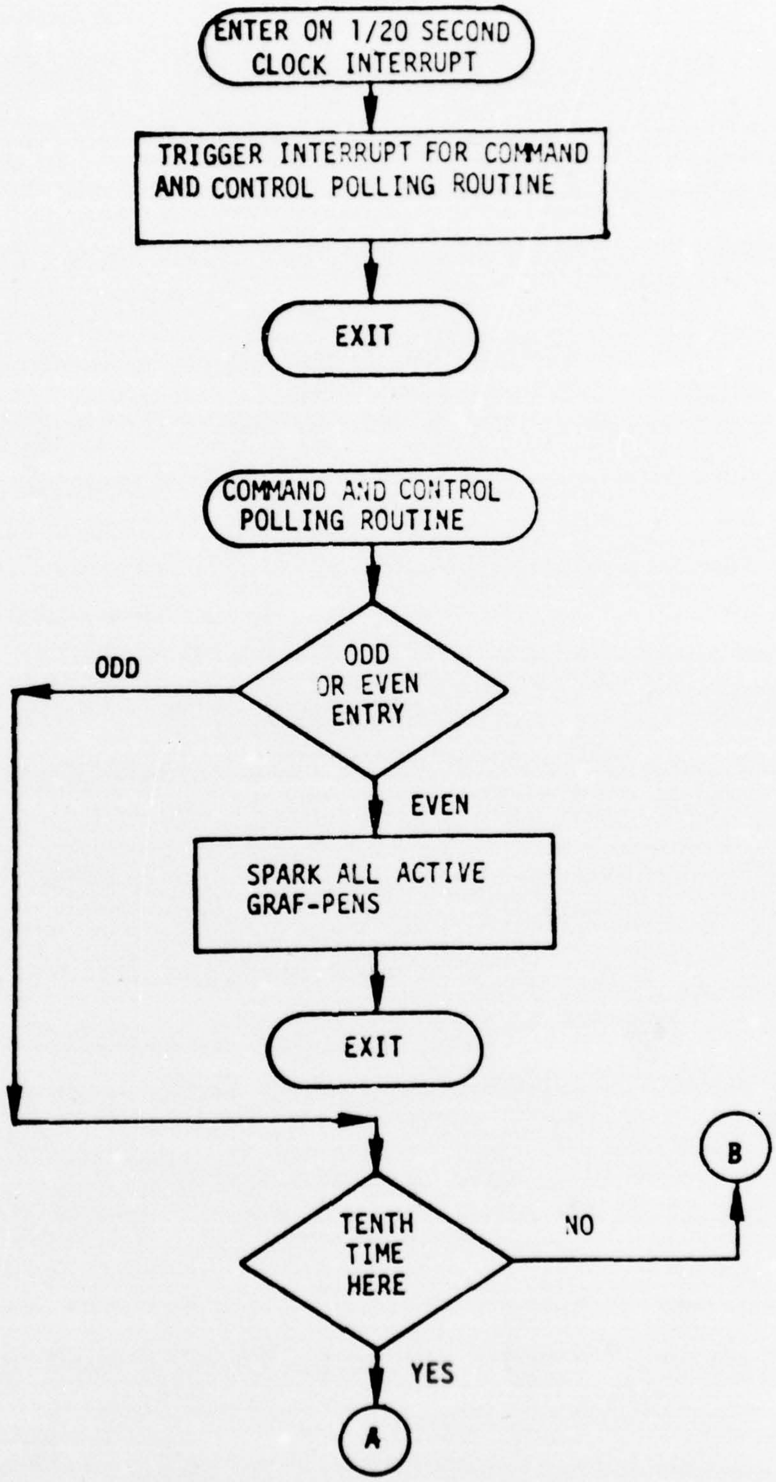


Figure 2-31. Overview of Command and Control Polling Routine

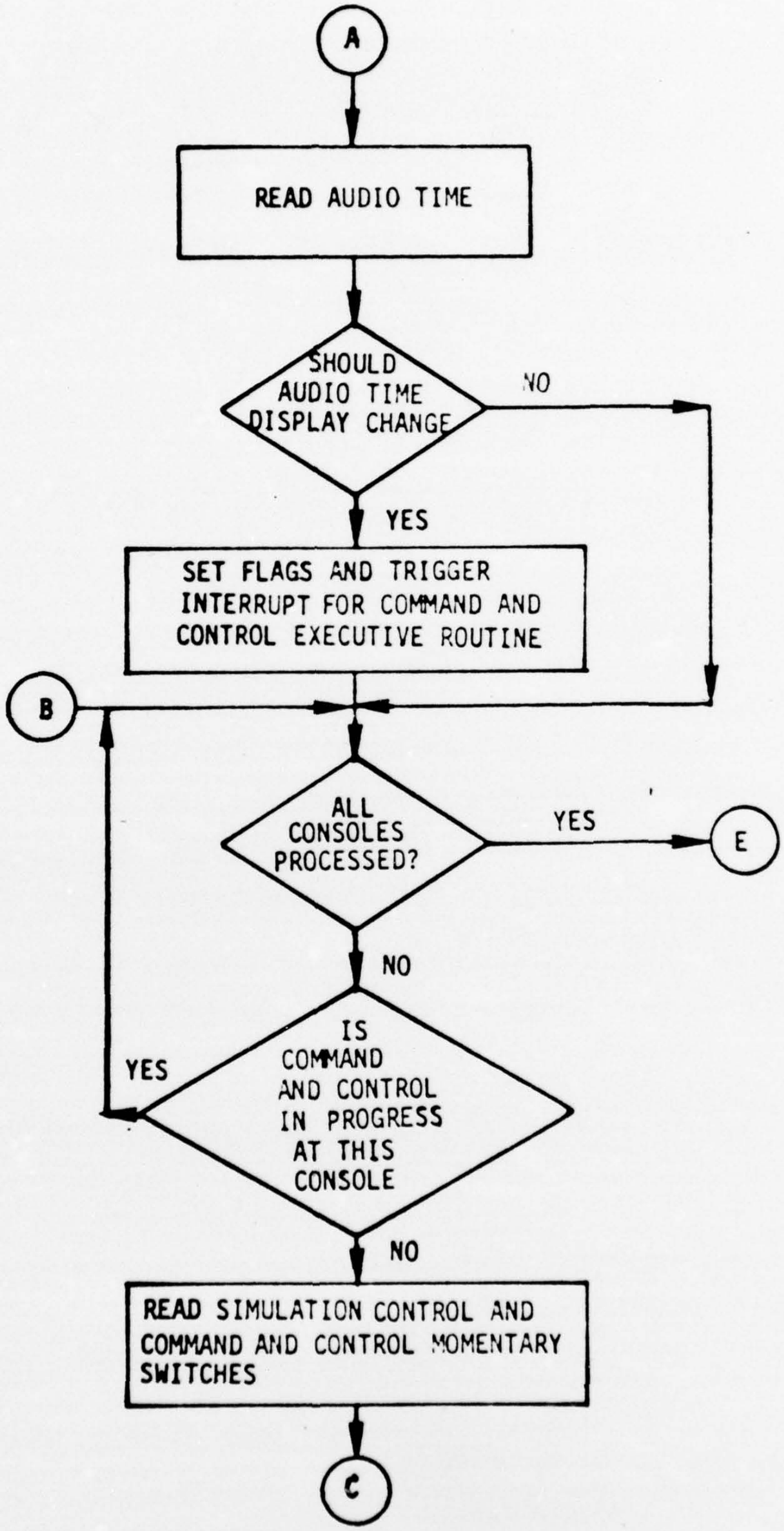


Figure 2-31. Overview of Command and Control Polling Routine, Cont'd.

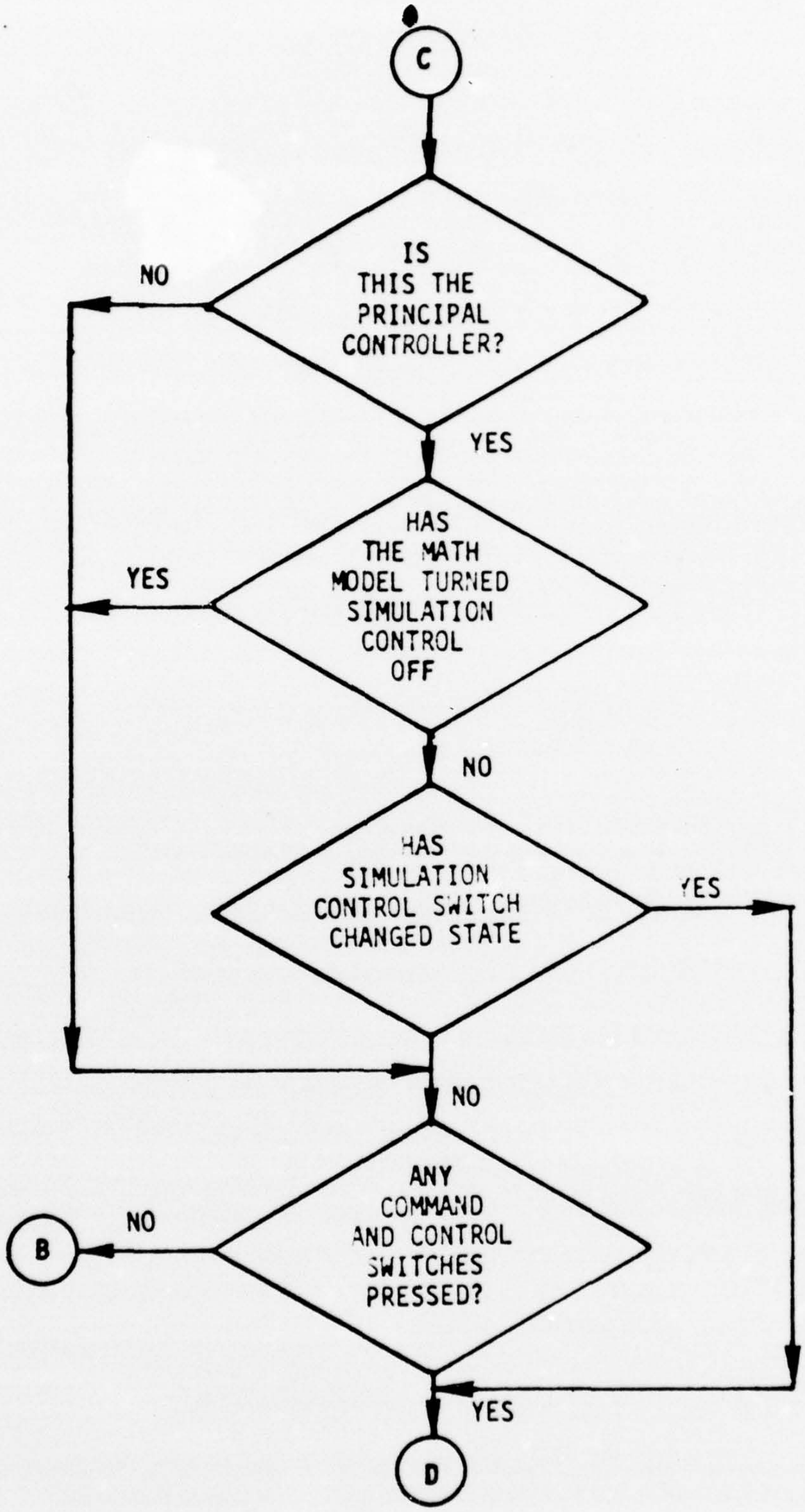


Figure 2-31. Overview of Command and Control Polling Routine, Cont'd.

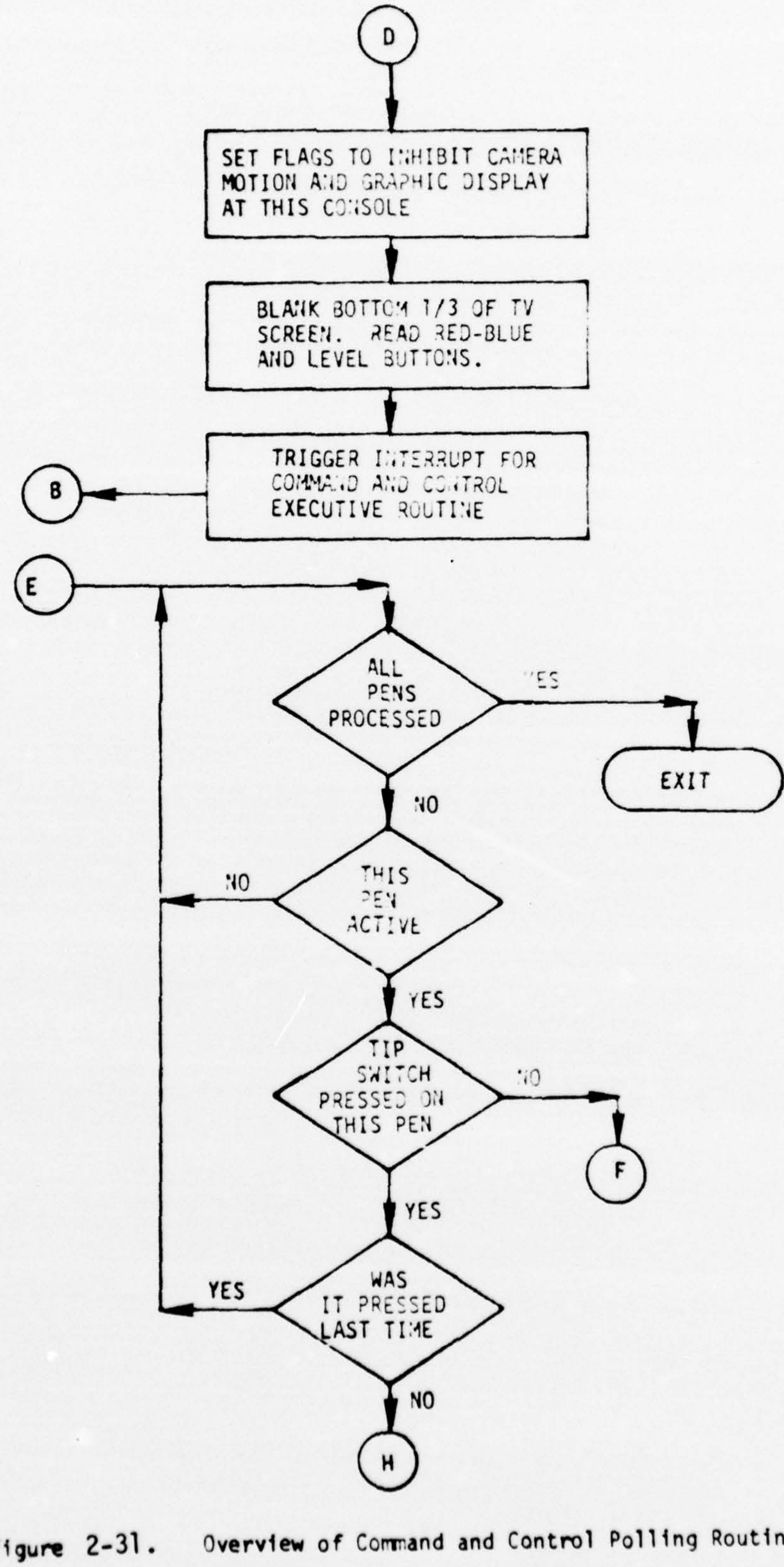


Figure 2-31. Overview of Command and Control Polling Routine, Cont'd.

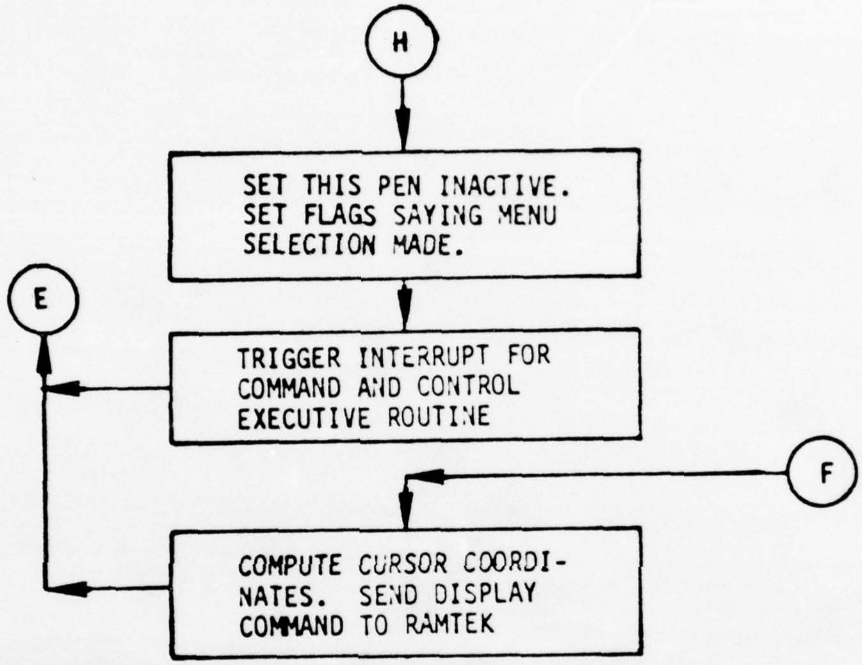


Figure 2-31. **Overview of Command and Control Polling Routine, Cont'd.**

- Read audio time encoder each twentieth entry. If time changes, trigger C&C executive.
- For each console at which a menu is not in progress, poll C&C switches on control panel and Ramalert buffer to see if a menu has been requested. If so:
  - -- inhibit camera motion at this console
  - -- inhibit graphic display at this console
  - -- blank bottom 1/3 of television screen using special effects generator
  - -- trigger C&C executive
- For principal controller console only, and only when math model has not inhibited simulation control, repeat above C&C steps for simulation control.

The command and control executive routine, on the other hand, runs at a lower priority interrupt level and performs the majority of processing for command and control. It creates the menu displays and reacts to menu selections made by the controllers. A flow diagram is shown in Figure 2-32. This portion of the program is heavily overlayed, as shown in Figure 2-33.

2.2.3.2.5 Ramtek Handler

The Ramtek is a hardware device which converts digital instruction input to an analog color viewo signal for display on television monitors. The Ramtek handler program processes all digital output from other computer programs to the Ramtek (with the exception of some cursor instructions transmitted directly from command and control). The video signals from the Ramtek are mixed with the map video signals from the cameras to create the images seen on the CATTS monitors - color military maps with color graphics overlays. See Figure 2-34 for an overview of Ramtek communications.

Ramtek displays are created by two programs - Graphic Display and Command and Control. Each is handled differently.

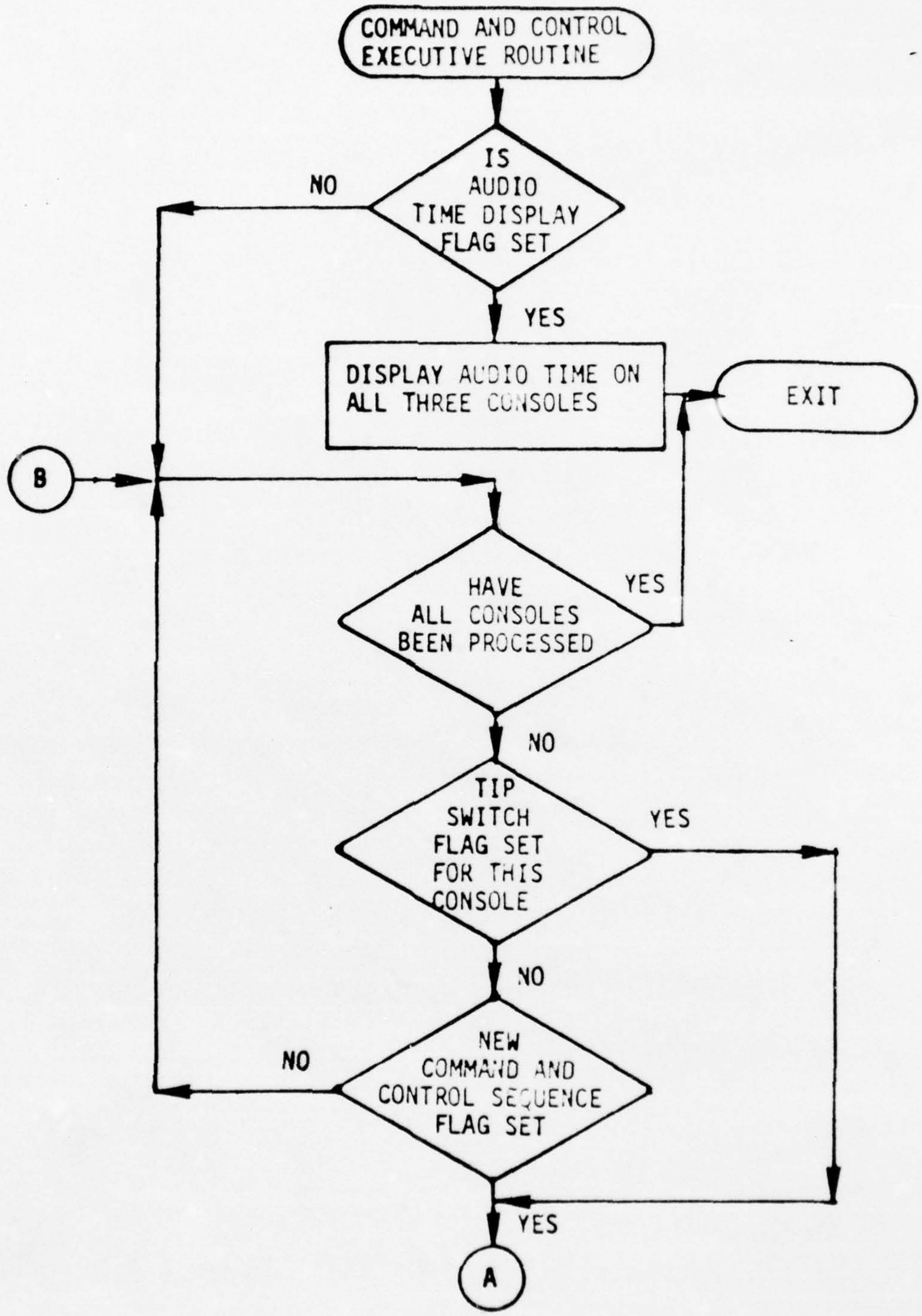


Figure 2-32. Overview of Command and Control Executive Routine

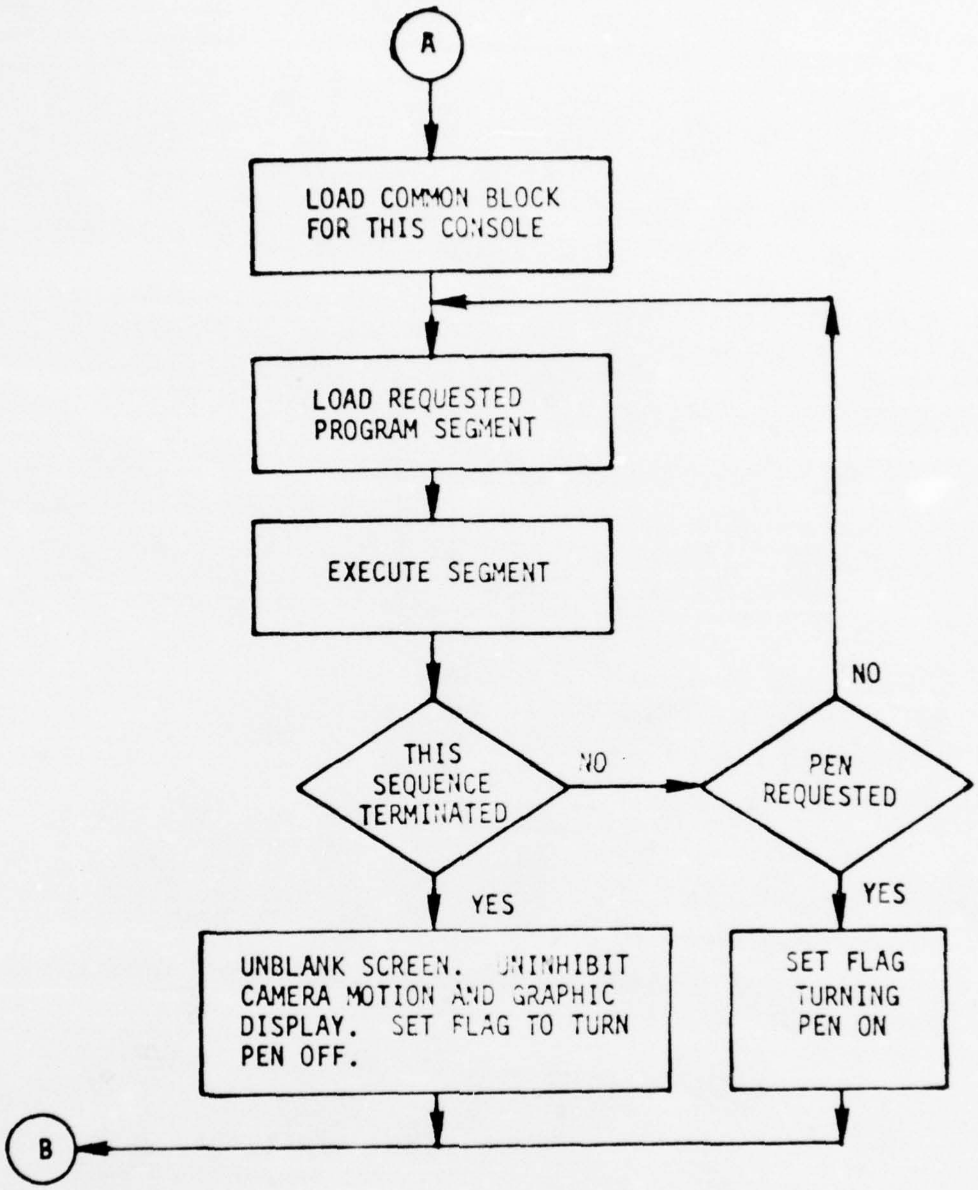


Figure 2-32. **Overview of Command and Control Executive Routine, Cont'd.**

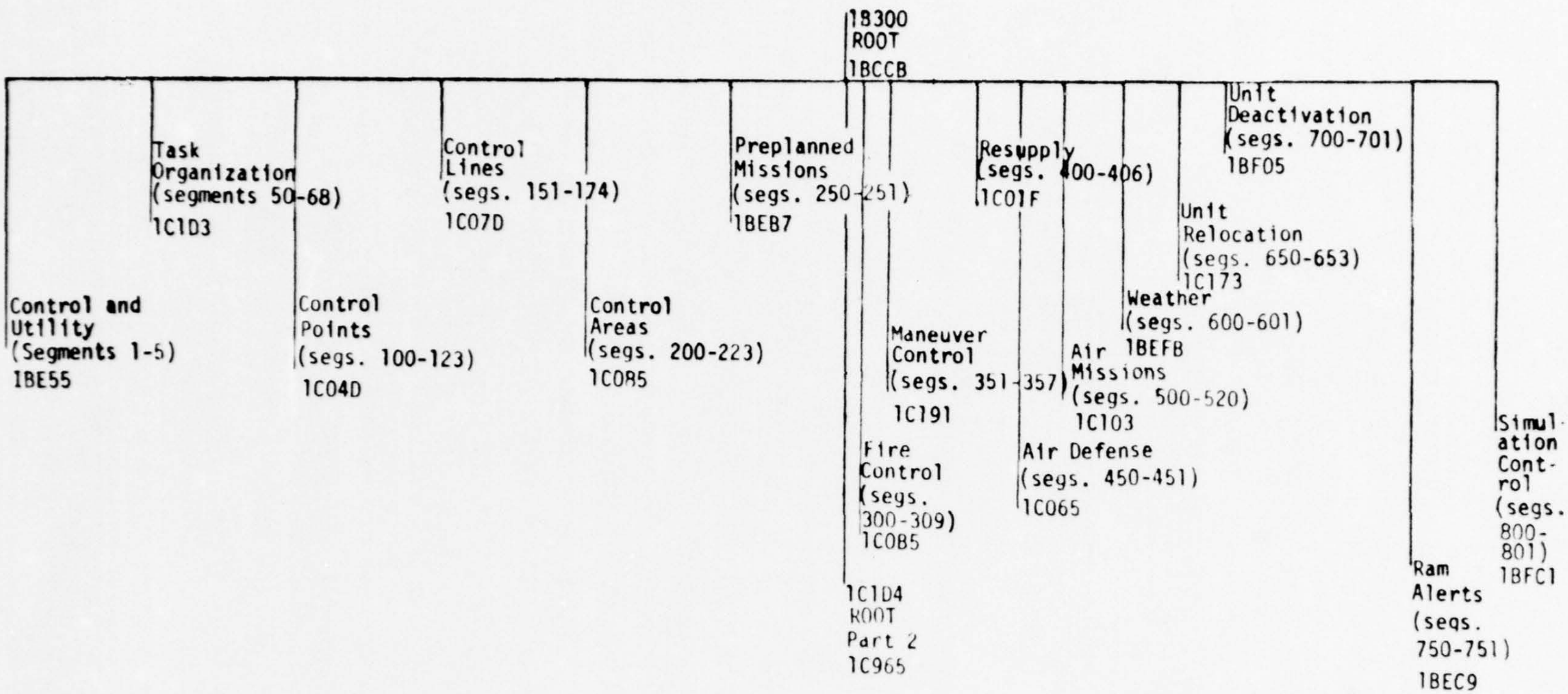


Figure 2-33. Command and Control Overlay Structure (Baseline 16A3-TS-01)

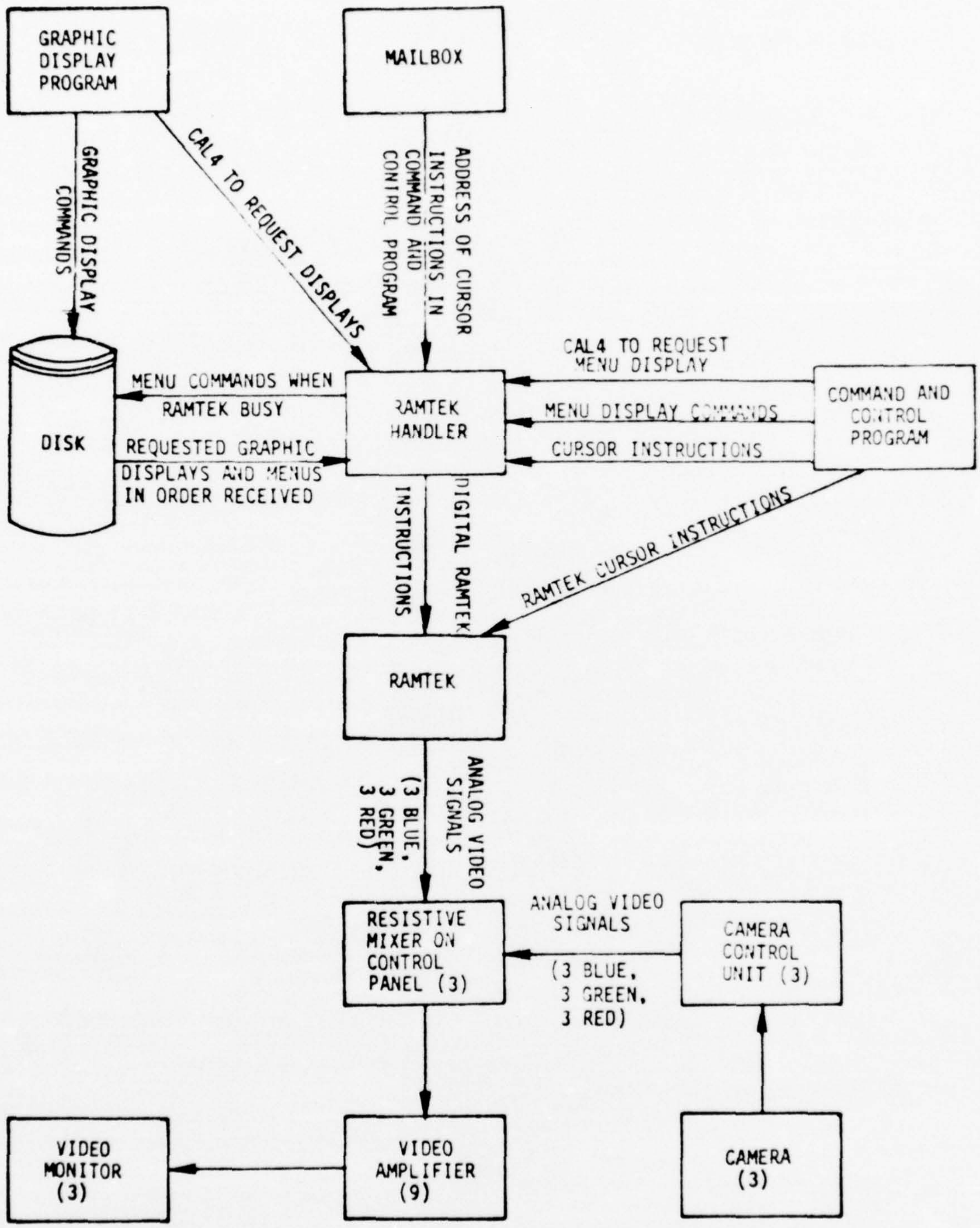


Figure 2-34. **Information Flow to/from Ramtek Handler**

Graphic Display computes each requested display and stores it in a fixed location on disk. Whenever it detects that a display should be redrawn, it does a CAL4 to tell the Ramtek handler to copy the appropriate disk sectors to the Ramtek. This scheme allows each separate display type to be computed only when absolutely necessary, such as when the camera is moved or when a math model timestep has occurred. When the display changes less drastically (for example, when a display type is deleted) no recomputation of displays is required.

Command and Control menu displays are requested by another CAL4. Whenever possible they are sent directly to the Ramtek. When necessary they are queued on a disk file. Queued Ramtek requests are serviced in the order received. Cursor instructions are written directly to the Ramtek by Command and Control whenever the Ramtek is not busy. Because it is often busy, the Ramtek handler always sends the last set of cursor instructions calculated by Command and Control as a prefix to every set of instructions it sends to the Ramtek. The address of these instructions is obtained through the Mailbox. When a cursor is not active, the instruction sent for that cursor is null.

3.3.2.6 Map Program

The map software is a program which is used to move the three cameras, convert camera encoder readings to usable information, and store this information in core storage. The addresses of the stored data are contained in the mailbox for access by Command and Control and Graphic Display. See Figure 2-35 for an overview of information flow to and from the map program.

The map program is clock driven at 1/10 second intervals. It polls the camera motion control switches at each console, and whenever it detects a change of state it sends the appropriate coded signal to the camera motion controller. This device drives the pan, tilt, and zoom motors for each camera. Whenever a camera stops moving, the map program waits one second for camera wobble to die down, then reads the pan, tilt, and zoom encoders plus the calibration data (from a disk file) for that camera, calculates a new set

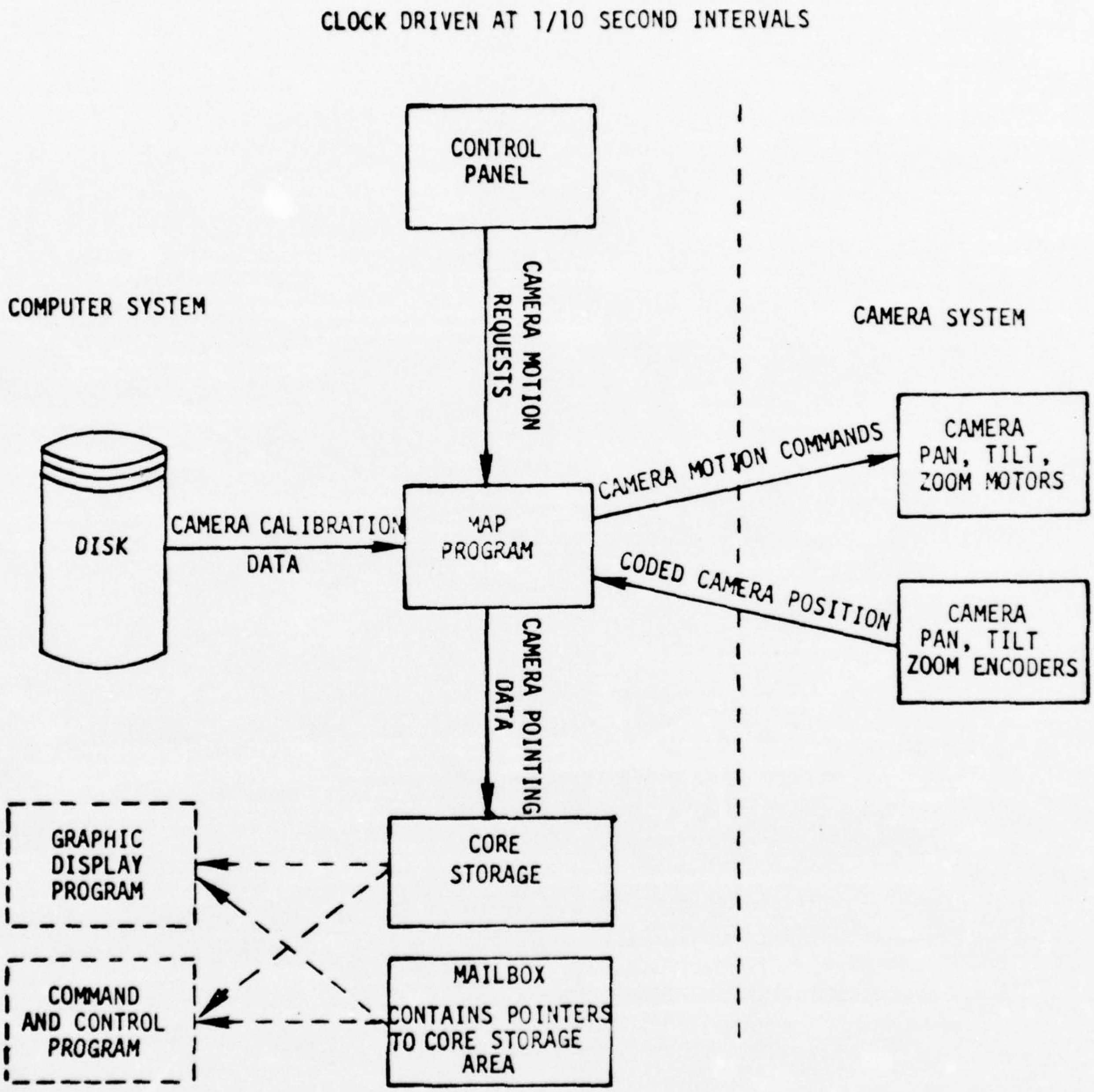


FIGURE 2-35. Simplified Overview of Information Flow in Map Program

of camera pointing data for the camera, and stores this data in a core location where it is accessed, through the mailbox, by Graphic Display and Command and Control. Figure 2-36 shows a high-level flow diagram of the map program.

3.3.2.7 Alert/RATT Program

The Alert/RATT program provides the software interface to the alphanumeric displays and the simulated RATT. It provides the controllers with a valuable, interactive information source. Figure 2-37 shows an overview of communications with other system elements. Capabilities provided include:

- Alert display. This is the normal operational mode. The alphanumeric alerts generated by the math model are queued up. As many as possible are displayed on the screen. See Figure 2-11 for a sample display. The remainder of the alerts are kept on file. The display does not change until the controller presses one of the special function switches on the alphanumeric keyboard, shown in Figure 2-38.
  - DROP - the top message on the screen id deleted. Queued messages move up until either there are no more or screen is full.
  - PRINT - the top message is first printed on the attached hand copy device, then dropped as in DROP.
  - SAVE - the top message is put in a special save file. These messages may be retrieved by using the SCAN ON/OFF key described below. They also reappear at the end of the normal queue.
  - SEND - the top message will be sent to another console with a one-line attached message. Software will request operator to key in both console id and message.
  - SCAN ON/OFF - displays the messages in the save file the first time it is pressed. The second time causes a reversion to normal processing.

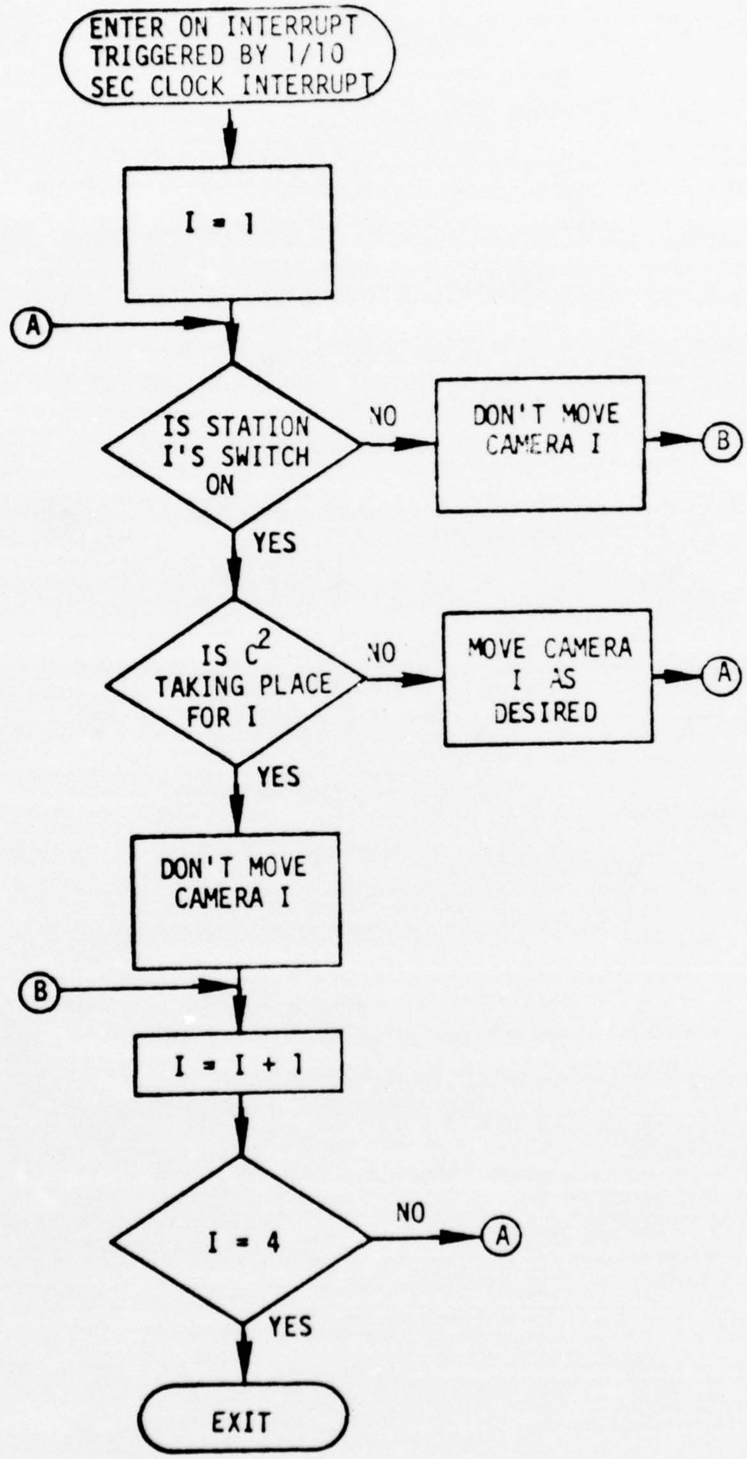


Figure 2-36. Map/Video Top-Level Flow Diagram (Baseline 16A3-TS-01)

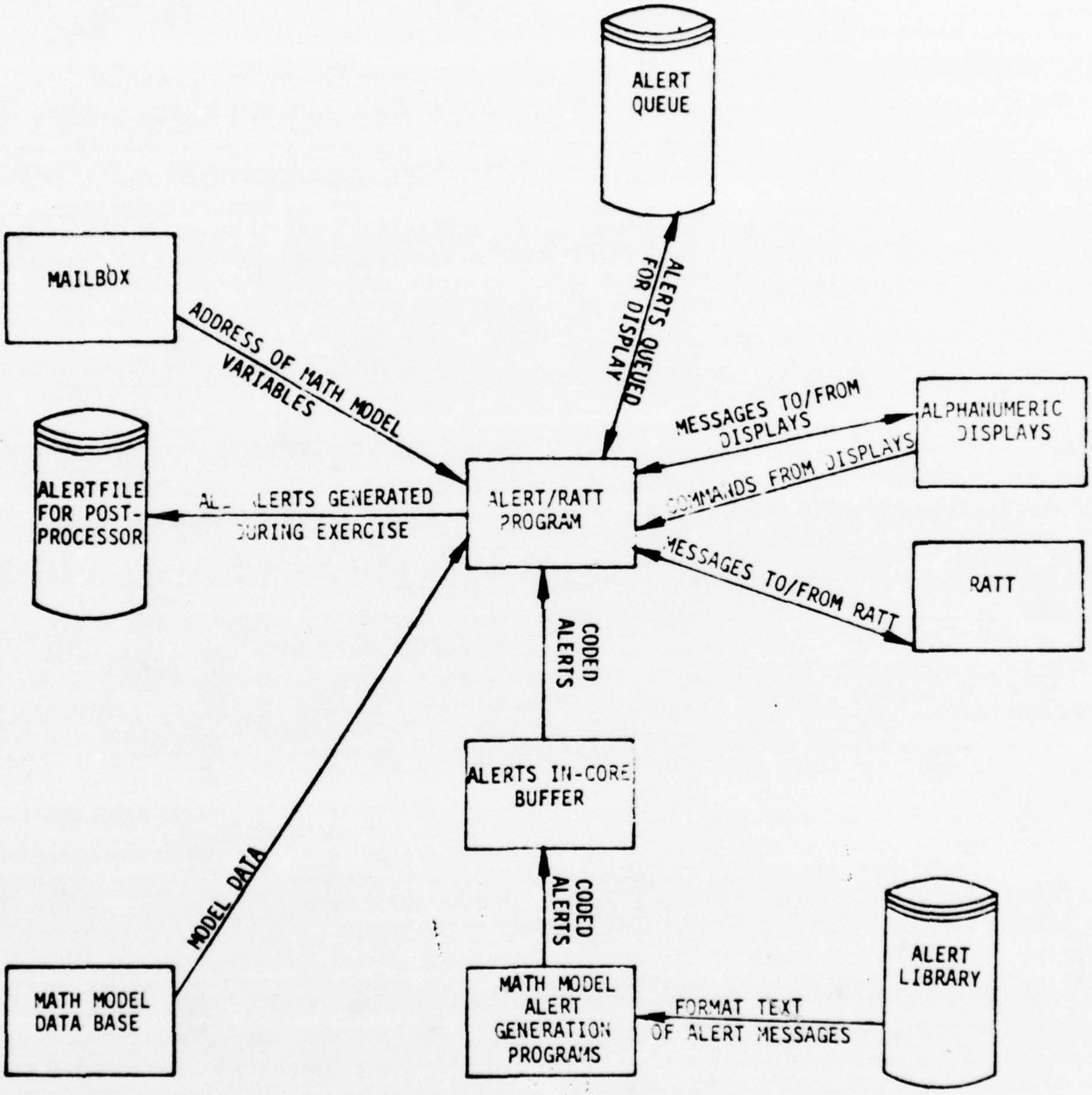


Figure 2-37. Overview of Information Flow for Alert/Ratt Program

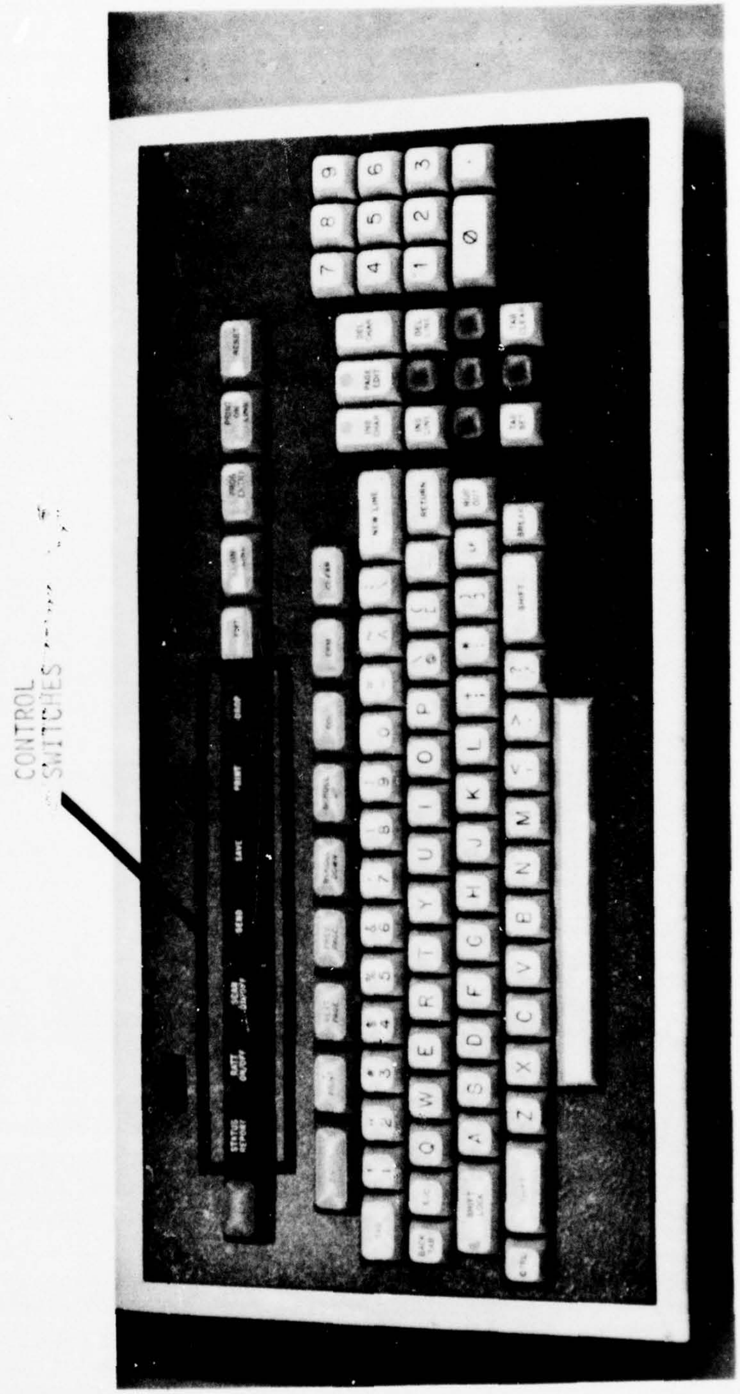


Figure 2-38. Special Function Keys Located on the Alphanumeric Display Keyboard

- RATT ON/OFF - allows the controller to send either a canned or a freshly composed message to the RATT. The software requests key-in of the appropriate commands and messages.
- STATUS REPORT - will cause the software to request a unit name key-in. A special status report for that unit will then be created and displayed until either PRINT or DROP is pressed. DROP reverts to normal processing. PRINT first prints the status report on the attached hard copy device and then reverts to normal processing.

Figure 2-39 is a high-level flow diagram of the operation of the Alerts Program, while Figure 2-40 shows the overlay structure.

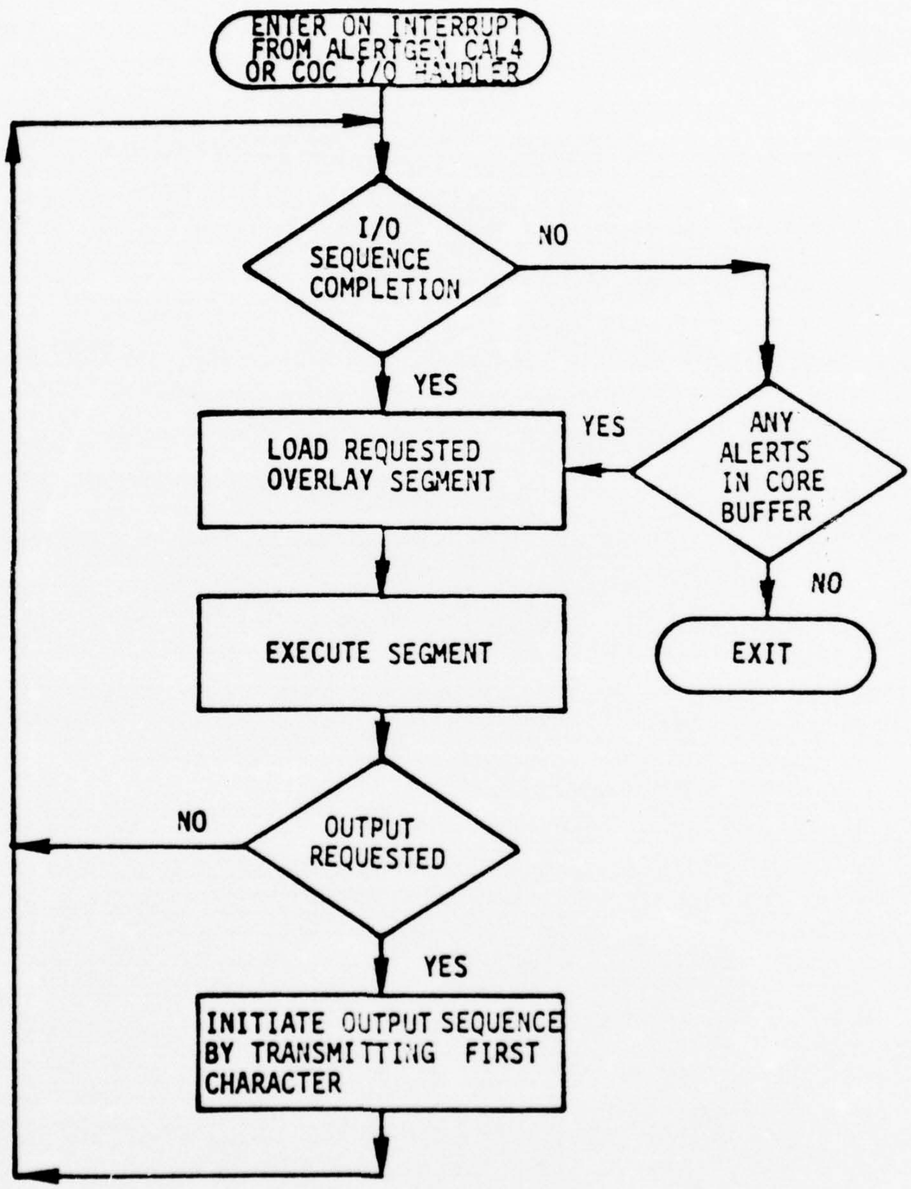


Figure 2-39. Overview of Alert/Ratt Program

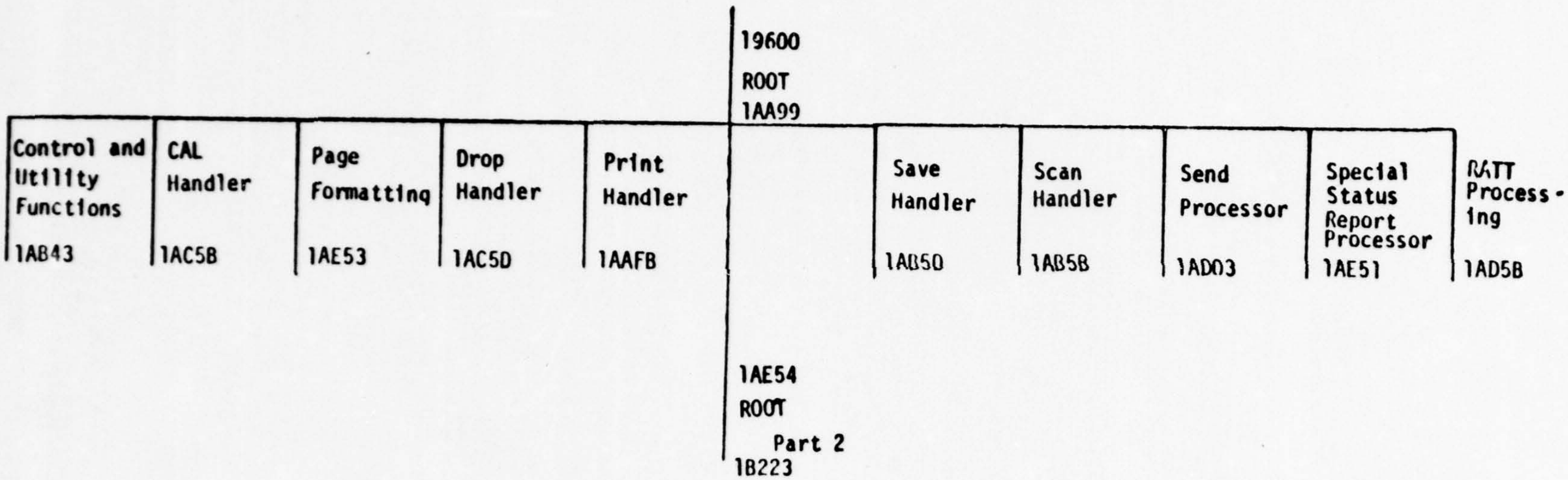


Figure 2-40. Alerts Overlay Structure (Baseline 16A3-TS-01)

3. DEFINITION OF TERMS

One of the problems for a user of CATTS is understanding the simulation concepts intrinsic to the mathematical model. An understanding of th se concepts is essential if the capabilities and limitations of the CATTS mathematical model are to be fully recognized by the user. Toward this end, it is necessary that the user completely understand the many "terms" that characterize the major concepts, entities, and data tables used by the CATTS mathematical model. In CATTS, "terms" correspond closely to the FORTRAN variables used in the mathematical model. In fact, to a great extent, the definition of a term is given by the meaning and usage of the one or more variables used to describe that term. For example, in CATTS a "bridge" is not represented by a single variable; instead it is a term defined by a set of numerical abstractions of some of the most important characteristics of a bridge. These numerical abstractions or "attributes" consist of such things as the X-Y coordinates of the endpoints of the bridge, the type of bridge (metal or pontoon), the water obstacle that the bridge crosses over, and the amount of damage, if any, against the bridge.

Thus, this section defines 163 of the terms that are used in the mathematical model. The term definitions are alphabetized and are defined in a standard format consisting of the term name, term description, and term reference, with each of these containing an English-language and programmatical definition for the benefit of the non-technical and technical user, respectively. The English-language term name is English, while the programmatical one would normally be a FORTRAN variable name. Similarly, the term description is presented, first, in English for the non-programming user, followed by a programmatical version largely oriented toward the programmer. The term reference also contains an English-language and a programmatical version. The English-language reference lists the English-language entity or entities to which the term applies or belongs, while the programmatical reference consists of the indexing or referencing of FORTRAN variables.

3.1 AIRCRAFT

3.1.1 English-language Definition

a) Term Name:

Aircraft

b) Term Description:

An aircraft is modeled in CATTS as an equipment whose equipment code has an indicator designating aircraft. When processing an equipment having an aircraft code, a large number of equipment attributes take on special meanings to represent the aircraft's performance characteristics. This data is used throughout the air module in creating a mission and moving an air unit over the battlefield.

c) Term References:

Aircraft is an entity in the model described by equipment attributes, many of which take on special meaning for aircraft. Aircraft are imbedded among ground equipments, and the attributes describing aircraft are referenced by equipment number. Aircraft data are used principally in creating an air mission, and moving it once it has been created.

3.1.2 Programmatical Definition

a) Term Name:

IEQCOD(IEQ) = -1

b) Term Description:

The following variables are data base input variables that are defined in the equipment input deck. They take on different meanings for aircraft, indicated by IEQCOD(IEQ) = -1 for equipment IEQ. Other equipment attributes not listed below and, therefore, not having special meaning for aircraft are listed under the term "equipment".

ROME(IEQ,1) = Maximum load aircraft can carry (pounds) at best modeled pressure density (PDBEST)

ROME(IEQ,2) = Maximum altitude of aircraft (meters)

ROME(IEQ,3) = Minimum speed of aircraft (meters/minute)

ROME(IEQ,4) = Cruise speed of aircraft (meters/minute)

ROME(IEQ,5) = Maximum speed of aircraft (meters/minute)

ROME(IEQ,6) = Maximum load aircraft can carry (pounds) at worst modeled pressure density (PDWORST)

<u>Note</u>: The correct negative value for ROME(IAG,6) will insure that an aircraft cannot fly at pressure densities below its capability.

ROME(IEQ,7) = Poorest meteorological visibility in which aircraft can continue its mission (meters)

ROFE(IEQ,1) = Fuel expenditure for losing altitude (lb/meter)

ROFE(IEQ,3) = Fuel expenditure at minimum speed, minimum load, best pressure density (lb/min)

ROFE(IEQ,4) = Same at cruise speed

ROFE(IEQ,5) = Same at maximum speed

ROFE(IEQ,6) = $\dfrac{\text{Fuel Expenditure rate at maximum load}}{\text{Fuel expenditure rate at minimum load}}$

ROFE(IEQ,7) = $\dfrac{\text{Fuel expenditure rate at worst pressure density}}{\text{Fuel expenditure rate at best pressure density}}$

ROFE(IEQ,8) = Not used

IAMTE(IEQ,1)= Takeoff delay time for this aircraft (minutes)

IAMTE(IEQ,2)= Landing delay time for this aircraft (minutes)

EQCAPAC(IEQ)= Fuel capacity of aircraft (pounds)

All of the above variables are input as part of the data base initialization, and are not changed thereafter.

c) Term References:

IEQCOD(IEQ) is indexed by equipment number, as are all other aircraft-specific variables listed above. The equipment index range from ($1 \leq \text{IEQ} \leq 80$). Aircraft are presently defined in the data base as equipment numbers 56 through 68, inclusive. A large number of subroutines use aircraft data, primarily AIREVENT in creating a new air mission, and AIRMOV in updating locations of air units each minute.

3.2 AIR DEFENSE

3.2.1 English-language Definition

a) Term Name:

Air Defense

b) Term Description:

When the air defense command and control action is taken, an indicator is set for those units selected on the menu to either:

1) fire at will
2) fire only if attacked
3) do not fire

This indicator is interpreted by the air defense function to either:

1) fire at all enemy aircraft within range and otherwise eligible for air defense fire
2) fire only at enemy aircraft attacking your specific unit determined by checking air unit target type value), if it is within range and otherwise eligible for air defense fire
3) do not fire air defense weapons under any circumstance

The air defense function is called each quarter-minute that a ground unit has detected an air unit. It determines eligibility of an air unit to air defense fire from each air defense weapon in the ground unit. An air unit is eligible if the ground unit's air defense flag is satisfied, and the ground unit has not fired air defense weapons this quarter minute. For each air defense weapon in the ground unit, each point of the air unit's movement during that quarter-minute (two end points plus any checkpoints between them) is examined pairwise. If it passes three checks; namely, 1) within altitude range of weapon, 2) within slant range of weapon, 3) detection occurred at the point, then the point is counted. If the other point of the pair is also counted, firing occurs for the whole interval of time. If just one point of a pair passes, the duration of firing over that pair of points is generated stochastically from a uniform distribution.

All pairs of points in the quarter-minute are examined this way, and the total duration of fire is computed. This duration is multiplied by the firing rate per gun, the number of such guns in the unit, and the suppression factor of the unit to yield the total number of rounds fired. Ammunition levels are reduced for the weapon fired.

c) Term References:

The air defense indicator array is an unit attribute, and is used exclusively in the processing of air defense weapon fire.

3.2.2 Programmatical Definition

a) Term Name:

IAIRDFLG(IU)

b) Term Description:

IAIRDFLG(IU) is a single-valued variable represented as an array. The array takes on the following variables:

1 = fire at will
2 = fire only if attacked
3 = do not fire

The array is initialized to fire at will, and can be changed only by interactively calling the air defense menu.

Subroutine AIRCAS performs the air defense processing during each time step. It is called by AIRMOV2 as air units are being moved, before any ground units have been processed in the time step. Having determined how many rounds an air defense weapon in a ground unit will fire at an air unit (see English term description), PRNTFIR is called to save appropriate data for the air defense fire report, FINFUN is called to find the weapons effects function used to compute possible air losses, and EFFNS performs the actual casualty computation. ADFALERT is called to generate appropriate air defense alerts.

c) Term References:

IAIRDFLG(IU) has a unit number index where ($1 \leq U \leq 100$). After initialization it is only set through command/control action in subroutine AIRDFLG. Subroutine AIRCAS is the air defense weapon processing routine.

3.3 AIR-GROUND DETECTION

3.3.1 English-language Definition

a) Term Name:

Air-Ground Detection

b) Term Description:

The air-ground detection table is a table of flags indicating whether or not detection has taken place for up to six different sensors on an air unit versus a ground unit. When an air mission is created, the six vectors for that specific air unit, one for each sensor, against each ground unit are cleared, indicating no detection of any ground unit by any sensor on the air unit. For a given air detection device, if an air unit-ground unit pair satisfy certain range, look angle, and altitude constraints, a probability of detection calculation is made and compared against a random number. If successful, the initial detection flag is set, an alert issued for that sensor, and the flag remains set until the air unit gets out of range of the sensor. See Section 5.3 for a discussion of the probability of detection calculation.

c) Term References:

The air-ground detection table is an attribute of 1) ground units, 2) air units, and 3) air sensors. It is used primarily by the target acquisition function to provide information to controllers regarding the location of opposing ground units.

3.3.2 Programmatical Definition

a) Term Name:

IAGDET(IU,IAU,IAS)

b) Term Description:

IAGDET(IU,IAU,IAS) is a single-valued variable represented as a bit matrix indicating whether ground unit IU has already been detected by air unit IAU with sensor IAS. IU is used to determine the first

index to the array. This table contains one bit for up to six air unit against each ground unit. If a bit is set (=1), a particular sensor in an air unit has detected a ground unit; otherwise (=0), detection has not occurred for that sensor. Each time a new sensor in an air unit detects a ground unit, a Superbee alert is generated. Table IAGDET contains 5,400 bits of air-ground detection information for up to 6 sensors on up to 10 air units for a maximum of 90 ground units (6*10*90=5,400). There are presently 90 ground units allowed in the model, since the 10 air units are imbedded in the unit arrays, making those 10 air units unavailable for ground unit definition.

c) Term References:

IAGDET is indexed by both ground and air unit numbers and sensor index number. The air unit number itself, IAU, is used as the second index, ($1 \leq IAU \leq 10$). The ground unit number is converted into a word/bit position combination. The word number index is determined as follows:

$$I = (IU-1)/32 + 1$$

The bit position is determined as follows, where MOD is the modulo function:

$$\text{bit position} = \text{MOD}(IU,32)$$

IU is the ground unit number in the above expressions, ranging from ($1 \leq IU \leq 100$). The word number is used as the first index, and the bit position is used to shift to the appropriate bit position within the word. The air sensor index ranges from ($1 \leq IAS \leq 6$), where

0 = Visual (eyeball)
1 = SLAR
2 = INFRARED
3 = KA30 camera
4 = KA60 camera
5 = VERT camera

The table is cleared (set to zero) by subroutine CLEARDET for all bit positions relating to a given sensor on an air unit. Subroutine AIRGRND updates the table each time step. A correspondence is extablished in a DATA statement in Subroutine AIRGRND relating the sensor types to actual equipment numbers, where much of the data describing the sensor is stored.

3.4 AIR MISSION

3.4.1 English-language Definition

a) Term Name:

Air Mission

b) Term Description:

An air mission consists of all the data specified in creating an air unit from the air mission command and control button.

1. air unit velocity at each check point (from 4 to 9 points must be defined)
2. air unit (X,Y,Z) at each check point
3. a color indicator for RED OR BLUE
4. an indicator for reconaissance or strike
5. a status code indicating not an active grount unit
6. a travel code indicating an airborne unit
7. the ordnance being delivered if a strike mission is specified
8. the check point designated as the target point for a strike
9. the air target type for a strike mission

As part of every event notice, the time of occurrence is tagged onto the air mission.

c) Term References:

Air missions are created via the command and control menu. The data defining the air mission is not changed without another command and control action to change the mission.

3.4.2 Programmatical Definition

a) Term Name:

IACPV(JCKPT,JAU)	ISTATU(IU)
IACPX(JCKPT,JAU)	ITRAV(IU)
IACPY(JCKPT,JAU)	NMPN(JAU)
IACPZ(JCKPT,JAU)	NTGT(JAU)
INOG(IU)	NTGTYP(JAU)
IOPSTU(IU)	

b) Term Description:

The following data is specified at the time an air mission is created through command and control:

IACPV (JCKPT,JAU)	Velocity at JCKPTth check point for air unit JAU.
IACPX (JCKPT,JAU)	X-coordinate of JCKPTth check point for air unit JAU.
IACPY (JCKPT,JAU)	Y-coordinate of JCKPTth check point for air unit JAU.
IACPZ (JCKPT,JAU)	Z-coordinate of JCKPTth check point for air unit JAU.
INOG(IU)	Operational grouping to which unit IU belongs. (If 0, unit IU does not belong to any op group) If IU is an air unit, INOG = 1 is a red air unit and INOG = - is a blue air unit.
IOPSTU (IU)	Operational state of unit IU. If IU is an air unit, IOPSTU = 1 if IU is on a recommaissance mission and = 2 if IU is on a strike mission.
ISTATU (IU)	Status code of unit IU: -1 -- air unit 1 -- in engagement, eligible for direct fire against enemy units in same engagement 0 -- other
ITRAV (IU)	Travel code of unit IU: 0 - not applicable 1 - unit avoids engagement if possible 2 - unit neither desires nor avoids engagement unit destination is reached 3 - unit seeks engagement 4 - air unit
NMPN (JAU)	The ordnance type being delivered by air unit JAU (JAU=1,10)

NTGTPT (JAU) — Index within /airoutes/ arrays of target for air unit JAU. = 0 if no target (recon mission).

NTGTYP (JAU) — Type of target for air unit JAU. = ground unit number or code for road, bridge, or X-Y point.

ITRAV (IU) is a model generated variable that is used by the Air Module; all other variables are interactively generated variables that are created from the air menu.

c) Term References:

The following arrays are indexed primarily by checkpoint number, JCKPT ($1 \leq JCKPT \leq 9$), where at least 4 check points must be defined, secondly by air unit number, IAU ($1 \leq IAU \leq 10$).

IACPV(JCKPT,IAU)
IACPX(JCKPT,IAU)
IACPY(JCKPT,IAU)
IACPZ(JCKPT,IAU)

The following arrays are indexed on air unit number alone, JAU($1 \leq JAU \leq 10$).

NMPN(JAU)
NTGT(JAU)
NTGTYP(JAU)

The following arrays are indexed by unit number, IU($1 \leq IU \leq 100$)

INOG(IU)
IOPSTU(IU)
ISTATU(IU)
ITRAV(IU)

Subroutine AIREVENT stores all air mission data in the appropriate arrays.

3.5 AIR MOVEMENT

3.5.1 English-language Definition

a) Term Name:

Air Movement

b) Term Description:

Movement of air units is accomplished by calculating an air route (see discussion of this term) for each active air unit once each time step. The air route consists of five to eight position points tagged with the relative time within the time step the air unit reached each position. Processing performed to accomplish air movement is as follows. Initially, the last position of the air unit in the previous time step is transferred to the first position this time step. Next, visibility, density altitude and fuel level are checked to determine if the mission is to be aborted (visibility) or if the air unit will crash (density altitude and fuel). An appropriate alert is generated for each case. Assuming that the air unit is continuing, the differential between current altitude and velocity and that specified at the next check point is computed. If the next checkpoint is a target unit which could be moving, the next checkpoint is corrected. The distance to the next checkpoint is compared with the distance to be covered to the end of the quarter minute at present speed to determine if a change in direction is required. If a checkpoint is reached first, time of arrival at the checkpoint is computed. The whole process of computing differentials for X, Y, Z to the next checkpoint is repeated for the entire one minute interval. Fuel used, which has been computed for each subinterval and accumulated, is removed from the air unit's supply.

c) Term References:

The air module is split into two parts. The air movement function comprises the second part, and is accomplished at the end of a time step. At the beginning of the next time step, the other part processes the position points computed by the air movement function.

3.5.2 Programmatical Definition

(Not applicable)

3.6 AIR ORDNANCE

3.6.1 English-language Definition

a) Term Name:

Air Ordnance

b) Term Description:

An air ordnance is modeled in CATTS as an equipment having an equipment code indicating air ordnance. This code causes processing by the air module, in particular the air-delivered weapons function, rather than by the ground equipment processing, which fires and moves ground equipment. The attributes of air ordnance types have the same programmatic names as ground equipments, but the data contained in many of them have special meaning for air ordnance. In particular, air ordnance weight and delivery constraints (altitude and speed) and other delivery parameters, types of aircraft equipped to carry the ordnance, and ordnance effectiveness data against various types of targets, are defined as attributes of air ordnance. These special air ordnance attributes are essential in creating an air strike mission and calculating the effects of the strike.

c) Term References:

Air ordnance is an entity in the model described by equipment attributes, many of which take on special meaning for air ordnance. Air ordnance types are imbedded among ground equipments, and the attributes describing them are referenced by equipment number. Air ordnance data is used principally in creating an air strike mission, and determining the effects of the strike when the ordnance is delivered.

3.6.2 Programmatical Definition

a) Term Name:

IEQCOD(IEQ) = -2

b) Term Description:

The following variables take on different meanings for air ordnance, indicated by IEQCOD(IEQ) = - 2 for equipment number IEQ. Other equipment attributes not listed below and, therefore, not having special meaning for air ordnance are listed under the term "equipment".

IPVCE(IEQ,J)	=	Equipment number of the Jth allowable aircraft type on which IEQ may be used. (Hence, a maximum of eight aircraft per equipment type.) The first zero encountered ⇒ no other allowable aircraft
IMINRE(IEQ)	=	Minimum range at which equipment may be used
IMAXRE(IEQ,1)	=	Maximum range at which equipment may be used
ROME(IEQ,1)	=	Weight of equipment (pounds) including standard ammunition load
ROME(IEQ,2)	=	Minimum aircraft speed (meters/minute) at which equipment can be used
ROME(IEQ,3)	=	Maximum speed at which equipment can be used (meters/minute)
ROME(IEQ,4)	=	Minimum altitude at which equipment can be used (meters)
ROME(IEQ,5)	=	Maximum altitude at which equipment can be used (meters)
ROME(IEQ,6)	=	Road crater radius, road type 1
ROME(IEQ,7)	=	Road crater radius, road type 2
ROMEIEQ,8)	=	Road crater radius, road type 3
ROFE(IEQ,1)	=	Fraction of personnel in personnel vulnerability Class 1 (standing) and within target area who are killed by this equipment
ROFE(IEQ,2)	=	Same for personnel vulnerability Class 2 (crouching)
ROFE(IEQ,3)	=	Same for personnel vulnerability Class 3 (prone)
ROFE(IEQ,4)	=	Fraction of equipment with IEQCLS = 1 and within target area which is damaged by this equipment
ROFE(IEQ,5)	=	Same for IEQCLS = 2
ROFE(IEQ,6)	=	Same for IEQCLS = 3
ROFE(IEQ,7)	=	Same for IEQCLS = 4
ROFE(IEQ,8)	=	Same for IEQCLS = 5
IAMTE(IEQ,1)	=	Number of ammunition types for this equipment, if any
IAMTE(IEQ,2)	=	Number of rounds in a standard load of that ammunition type

IAMTE(IEQ,3) = Rate of fire of this equipment (rounds/minutes)

IAMTE(IEQ,4) = Number of drops per pass (applies to bombs)

IAMTE(IEQ,5) = Distance between drops (meters)

IAMTE(IEQ,6) = Dud probability

IAMTE(IEQ,7) = Kill probability *100 against bridge type 1

IAMTE(IEQ,8) = Kill probability *100 against bridge type 2

IEQCLS(IEQ)	=	1	IEQ is a 250-lb low-drag bomb
	=	2	IEQ is 500-lb low-drag bomb
	=	3	IEQ is 750-lb low-drag bomb
	=	4	IEQ is 1000-lb low-drag bomb
	=	5	IEQ is 2000-lb low-drag bomb
	=	6	IEQ is 250-lb high-drag bomb
	=	7	IEQ is 500-lb high-drag bomb
	=	8	IEQ is 750-lb high-drag bomb
	=	9	IEQ is Maverick
	=	10	IEQ is Shrike
	=	11	IEQ is 2.75 rockets
	=	12	IEQ is 20-mm cannon
	=	13	IEQ is CBU-2
	=	14	IEQ is CBU-24
	=	15	IEQ is Rockeye
	=	16	IEQ is Napalm

IEQCLS(IEQ) is a data base input variable that is defined in the namelist input deck; all other variables are data base input variables that are defined in the equipment input deck.

c) Term References:

IEQCOD(IEQ) is indexed by equipment number, as are all other aircraft-specific variables listed above. The equipment index range from ($1 \leq IEQ \leq 80$). Air ordnance types are presently defined in the data bases as equipment numbers 22 through 25 and 69 through 80, inclusive. A large number of subroutines rise air ordnance data, primarily AIREVENT in creating an air strike mission, and ADW in computing ordnance effectiveness against various types of ground targets.

3.7 AIR ROUTE

3.7.1 English-language Definition

a) Term Name:

Air Route

b) Term Description:

An air route is computed each minute for each active air unit. The route consists of from 5 to 8 points consisting of the following information for a given air unit:

1. X,Y position
2. altitude
3. time (within the time step)

Each time step, the air unit route data is calculated for the beginning and end of the time step, and at each quarter time step interval. A maximum of three check points can be defined within the time step, to yield a maximum of eight route points.

Air units are moved along these route points, and displayed only on the quarter time step interval. Detection and air defense processing is based on the air route position data and time tag.

c) Term References:

All air route data values are air unit attributes. They are calculated each time step as the air unit oves along its assigned flight path.

3.7.2 Programmatical Definition

a) Term Name:

IXAIR(JAIRTE,JAU)
IXAIRFG(JAIRTE,JAU)
IYAIR(JAIRTE,JAU)
IYAIRFG(JAIRTE,JAU)
IZAIR(JAIRTE,JAU)
IZAIRFG(JAIRTE,JAU)
TIMXY(JAIRTE,JAU)

b) Term Description:

The air unit arrays defined below are model generated variables that are used in the Air Module. They combine to describe the air route of air unit JAU in a given time step. The route consists of a minimum of four and a maximum of eight points.

Four duplicate arrays are defined solely for use by the foreground graphic display function so that the foreground sees static information. Data in arrays IXAIR,IYAIR,IZAIR, and TIMXY is dynamic in nature, and therefore, not suitable for display purposes.

IXAIR (JAIRTE,JAU) X-coordinate of air unit JAU at JAIRTEth point in this minute.

IXAIRFG (JAIRTE,JAU)

X-coordinate of air unit JAU (1 < = JAU < = NLDU-NFDU+1) at the JAIRTEth point (1 < = JAIRTE < = 8) in this minute. The minute is broken down into quarter min. points plus any air-route points that are arrived at in the minute interval. This array is identical to IXAIR except IXAIR represents current minute data. IXAIRFG exist only for use by the foreground programs.

IYAIR (JAIRTE,JAU) Y-coordinate of air unit JAU at JAIRTEth point in this minute.

IYAIRFG (JAIRTE,JAU)

Y-coordinate of air unit JAU (1 < = JAU < = NLDU-NFDU) at the JAIRTEth point (1 < = JAIRTE < = 8) in this minute.
(see def. of IXAIRFG.)

IZAIR (JAIRTE,JAU) Z-coordinate of air unit JAU at JAIRTEth point in this minute.

IZAIRFG (JAIRTE,JAU)

Z-coordinate of air unit JAU (1 < = JAU < = NLDU-NFDU) the JAIRTEth point (1 < = JAIRTE < = 8) in this minute. (see def. of IXAIRFG.)

TIMXY (JAIRTE,JAU) Time at which air unit JAU reaches JAIRTEth point in this minute.

TIMXYFG (JAIRTE,JAU)
Time at which air unit JAU reached JAIRTEth point in preceding minute. Used by the foreground programs. (see IXAIRFG).

c) Term References:

All of the above arrays are indexed primarily by air route index, JAIRTE ($1 \leq$ JAIRTE ≤ 8), secondly by air unit number JAU)$1 \leq$ JAU ≤ 10). Subroutine AIRMOV calculates air unit positions and associated times each minute for arrays IXAIR,IYAIR,IZAIR,TIMXY. Subroutine FORMAIN copies these values into the static arrays IXAIRFG,IYAIRFG, IZAIRFG, and TIMXYEG for foreground graphic display processing

3.8 AIR SENSOR

3.8.1 English-language Definition

a) Term Name:

Air Sensor

b) Term Description:

An air sensor is modeled in CATTS as an equipment having an equipment code indicating air sensor. This code causes processing by the air module, in particular the air-ground target acquisition function, rather than by the ground equipment processing. The attributes of air sensors have identical programmatic names as ground equipments, but the data contained in many of them have special meaning for air sensors. In particular, air sensors have a weight and list of allowable aircraft that are factors when a new air mission is created, and range, altitude, and speed constraints when the sensor is evaluated for ground target acquisitions.

c) Term References:

Air sensor is an entity in the model described by equipment attributes, many of which take on special meaning for air sensors. Air sensors are imbedded among equipments, and the attributes describing air sensors are referenced by equipment number. Air sensor data are used principally in creating an air mission, and detecting ground targets once the air unit has taken off.

3.8.2 Programmatical Definition

a) Term Name:

IEQCOD(IEQ) = -3

b) Term Description:

The following variables are data base input variables that are defined in the equipment deck. They take on different meanings for air sensors, indicated by IEQCOD(IEQ) = - 3 for equipment number IEQ. Other equipment attributes not listed below and, therefore, not having special meaning for air sensors are listed under the term "equipment".

IPVCE(IEQ,J)	=	Equipment numbers of the Jth allowable aircraft type on which IEQ may be used. (Hence, a maximum of eight aircraft per equipment type.) The first zero encountered ⇒ no other allowable aircraft
IMINRE(IEQ)	=	Minimum range at which equipment may be used
IMAXRE(IEQ,1)	=	Maximum range at which equipment may be used
ROME(IEQ,1)	=	Weight of equipment (pounds) including standard ammunition load
ROME(IEQ,2)	=	Minimum aircraft speed (meters/minute) at which equipment can be used
ROME(IEQ,3)	=	Maximum speed at which equipment can be used (meters/minute)
ROME(IEQ,4)	=	Minimum altitude at which equipment can be used (meters)
ROME(IEQ,5)	=	Maximum altitude at which equipment can be used (meters)
ROME(IEQ,6)	=	Not used
ROME(IEQ,7)	=	Not used
ROME(IEQ,8)	=	Not used

c) **Term References:**

IEQCOD(IEQ) is indexed by equipment number, as are all other aircraft-specific variables listed above. The equipment index range from ($1 \leq IEQ \leq 80$). Air sensors are presently defined in the data base as equipment numbers 51 through 55, inclusive. The principal subroutines using air sensors are AIREVENT in creating a new air mission, and AIRGRND in determining ground target acquisitions by air units.

3.9 AIR TARGET TYPE

3.9.1 English-language Definition

a) Term Name:

Air Target Type

b) Term Description:

An air target type is selected for exactly one point on an air strike mission. The selection is made under the heading "Target Type" on the first page of the air menu. Choices on this heading are among the following:

1) Area target -- point designated by graph pen is used as target point
2) Hard target -- designated point is corrected to nearest unit, then used as target point
3) Bridge -- designated point is corrected to nearest bridge center, then used as target point
4) Road -- designated point is corrected to nearest road center, then used as target point

Ordnance delivered against an area target is treated as though no specific equipment in any unit was attacked. A lethal area is built around the point of impact and any unit area intersecting that lethal area is assessed for casualties based on the proportion of the unit affected. Ordnance delivered against a hard target is treated as though specific ground equipments were attacked in the designated unit, and a table of kill probabilities for the ordnance against each ground equipment is used to evaluate specific equipment damage on a priority basis. The area target logic is then used to account for additional casualties to simulate effects in the area surrounding the point of impact.

Ordnance delivered against a bridge is handled by computing the impact point, and determining if the impact point, and determining if the impact point is on the bridge itself. If so, a damage probability is assessed via table look-up.

Ordnance delivered against a road is handled by computing the compact point and a crater around that point. If the crater intersects with the road segment, damage is assessed.

c) Term References:

Attribute of air mission, indexed by air mission number.

3.9.2 Programmatical Definition

a) Term Name:

NTGTYP(IAU)

b) Term Description

NTGTYP(IAU) is an interactively generated variable created from the air menu. It is a target type indicator for an air unit on a strike mission, taking on the following values from the air menu

0 = not an air strike
1 - 100 = hard target unit number
201 = area target indicator
202 = road indicator
203 = bridge indicator

c) Term References:

Indexed by air mission number, IAU ($1 \leq IAU \leq 10$). The following subroutines use NTGTYP:

ADW
AIREVENT
AIRMOV
AIRMOV2
ORDPRI

3.10 AIR UNIT

3.10.1 English-language Definition

a) Term Name:

Air Unit

b) Term Description

An air unit is created through the air mission command and control function. All data listed under the term "air mission" also applies to an air unit. In addition, during an air unit's processing in a time step, a large number of data is calculated and stored to indicate from one subroutine to another where the air unit is, whether or not it has been detected, whether or not it has detected any ground units, etc. These arrays are defined in the programmatic description.

c) Term References:

When an air mission is created a large number of static variables are filled with permanent information, not altered from time step to time step. These are described under the term "air mission". Those additional data shown for "air unit" and not for "air mission" are dynamic data recalculated each time step. All data listed here are air unit attributes.

3.10.2 Programmatical Definition

a) Term Name:

(See list of program variables under term description)

b) Term Description:

The complete list of all variables defining an air unit follows:

AIRBORT (JAU)	Logical variable to indicate if an air unit mission has been aborted =.TRUE. If aborted ; =.FALSE. If not ; 1 < = JAU < = 10
DTCTATPT (JAU)	= .TRUE. if air unit was detected at point JAU. (JAU is index to /AIRLOC/ arrays.)
IACPV (JCKPT,JAU)	Velocity at JCKPTth check point for air unit JAU
IACPX (JCKPT,JAU)	X-coordinate of JCKPTth check point for air unit JAU.

IACPY (JCKPT,JAU) X-coordinate of JCKPTth check point for air unit JAU.

IACPZ (JCKPT,JAU) Z-coordinate of JCKPTth check point for air unit JAU.

IAGDET (JU,JAU,JAS) Bit matrix indicating whether ground unit IGU has already been detected by air unit JAU with sensor JAS. IGU is used to determine JU.

IGADET (JU,JAU) Bit matrix indicating whether ground unit IGU has already detected air unit JAU. IGU is used to determine JU.

INOG (IU) Operational grouping to which unit IU belongs. (If 0, unit IU does not belong to any op. group) If IU is an air unit, INOG = 1 is a red air unit and INOG = 2 is a blue air unit.

IOPSTU (IU) Operational state of unit IU. If IU is an air unit, IOPSTU = 1 if IU is on a recommaissance mission and = 2 if IU is on a strike mission.

IPRVENG (JU,JAU) Bit matrix indicating whether ground unit IGU is currently engaging air unit JAU with air defense weapons. IGU is used to determine JU.

ISTATU (IU) Status code of unit IU:

- -1 -- air unit
- 1 -- in engagement, eligible for direct fire against enemy units in same engagement.
- 0 -- other

ITRAV (IU) Travel code of unit IU:

- 0 - not applicable
- 1 - unit avoids engagement if possible
- 2 - unit neither desires nor avoids engagement until destination is reached
- 3 - unit seeks engagement
- 4 - air unit

IXAIR (JAIRTE,JAU) X-coordinate of air unit JAU at JAIRTEth point in this minute.

IXAIRFG (JAIRTE,JAU)
X-coordinate of air unit JAU (1 < = JAU < =NLDU-NFDU+1) at the JAIRTEth point (1 < = JAIRTE < = 8) in this minute. The minute is broken down into quarter min. points plus any air-route points that are arrived at in the minute interval. This array is identical to IXAIR except IXAIR represents current minute data. IXAIRFG exist only for use by the foreground programs.

IYAIR (JAIRTE,JAU) Y-coordinate of air unit JAU at JAIRTEth point in this minute.

IYAIRFG (JAIRTE,JAU)
Y-coordinate of air unit JAU (1 < = JAU < = NLDU-NFDU) at the JAIRTEth point (1 < = JAIRTE < = 8) in this minute.
(see def. of IXAIRFG.)

IZAIR (JAIRTE,JAU) Z-coordinate of air unit JAU (1 < = JAU < = NLDU-NFDU) the JAIRTEth point (1 < = JAIRTE < = 8) in this minute. (see def. of IXAIRFG.)

JIMSFLAG(JAU) Logical array used to indicate if air mission JAU has been aborted (JAU=1,10)(Same variable as AIRBORT).

JTGTPT (JAU) Index within /AIRLOC/ arrays for target point for air unit JAU. =0 if air unit JAU has no target during this minute.

KAIRPTS (JAU) Number of points calculated for air unit JAU during preceding minute. Used by the foreground.

NAIRPTS (JAU) Number of points calculated for air unit JAU during this minute.

NEXTACP (JAU) Index within /AIROUTES/ arrays of next check point for air unit JAU (1 < = JAU < = 10).

NMPN (JAU) The ordnance type being delivered by air unit JAU (JAU=1,10)

NTGTPT (JAU) Index within /AIROUTES/ arrays of target for air unit JAU. =0 if no target (recon mission).

TIMXY (JAIRTE,JAU) Time at which air unit JAU reaches JAIRTEth point in this minute.

TIMXYFG (JAIRTE,JAU) Time at which air unit JAU reached JAIRTEth point in preceding minute. Used by the foreground programs. (see IXAIRFG).

AIRBORT, IACPV, IACPX, IACPY, IACPZ, INOG, and IOPSTU are interactively generated variables that are created from the air menu; all other variables are model generated variables that are used in the Air Module.

c) Term References:

All arrays defined above with the index IAU or JAU are indexed on air unit ($1 \leq$ IAU,JAU ≤ 10). All arrays with the unit number index, IU range from ($1 \leq$ IU ≤ 100). The index JAIRTE above ranges from ($1 \leq$ JAIRTE ≤ 8) for a maximum of 8 air position points within a single time step. The index JCKPT above ranges from ($1 \leq$ JCKPT ≤ 9) for up to 9 check points allowed when an air unit is created. A large number of subroutines, primarily in the Air Module, use above variables.

3.11 ALERTS

3.11.1 English-language Definition

a) Term Name:

Alerts

b) Term Description:

Alerts are text messages which may be displayed in one or both of the following places: They may either be transmitted to the controllers from the math model via the alphanumeric display located at each of the controller's stations or they may be stored on disk during the execution of a game and retrieved at the end of the game as part of the post game summary. Alerts are generated to indicate changes in the tactical situation and conditions which may impact on the tactical situation. Alerts are used to indicate the following types of tactical changes and conditions:

1. Ground unit firing at air unit
2. Air unit delivering ordinance at ground unit
3. Casualties of ground units due to air strikes
4. Air units command rejected due to low fuel
5. Air unit low on ordinance
6. Air unit changing mission
7. Air mission command accepted
8. Cancellation of air missions
9. Air unit detecting ground unit
10. Air unit casualties due to ground fire
11. Air unit low visibility
12. Air unit crashed
13. Air unit landed
14. Fire control measure violations
15. Units crossing designated control measures
16. Aural detection
17. Visual detection
18. Detection with radar
19. Unattended ground sensor alarms

20. Road damage
21. Bridge damage
22. Ground unit receiving fire from firing at ground unit
23. Casualties from ground fire
24. End of receiving fire from ground unit
25. Low fuel levels
26. Change in speed
27. Encounters of obstacles
28. Encounters of minefields
29. Low ammunition levels
30. Change in REDCON
31. Resupply acknowledgements
32. Restart of model
33. Reinitialization of model
34. Support fire at X,Y location
35. Destruction of command and control headquarters
36. Weather changes

c) Term References:

The alerts type number is used to associate the data to be displayed with an alert with the appropriate format for the alert text message. Alert messages are generated by several modules within the math model. The air module uses alerts to indicate the status of air units and the damage caused by air units. The control measure module indicates violations of control measures through alert messages. The detection modules indicate aural, visual and sensor detections by specified message types. The firing module displays alerts for each type of damage that is inflicted by ground based weapons. The simulation control module indicates all simulation control that is activated except "freeze" during a game. The movement module displays changes of movement as they occur. Encounters with obstacles are indicated by alerts by the obstacle module. Weather changes are displayed via weather alerts.

3.11.2 Programmatical Definition

a) Term Name:

(Alerts are strings of characters representing the information discussed below and are not represented by any single variable.)

b) Term Description:

Each alert message is defined by a message type and information within the message. The following is a list of all alert message types along with a brief description of the condition for generating each type. An explanation of each type message is also provided.

Type 59: Ground unit,location,time FIRING AT AIR UNIT air unit

Condition: When ground unit starts firing its air defense weapon at air unit.

Explanation: Self-explanatory

Type 70: Ground unit,location,time CEASING FIRE AT AIR UNIT unit name

Condition: Ground unit was firing at air unit and has stopped firing at it.

Explanation: Self-explanatory

Type 71 Air unit, location, time DELIVERED air ordnance AT location AGAINST target

Condition: Air unit delivers ordnance.

Explanation: Air unit has delivered an air ordnance at ground unit, a road, a X-Y point or a bridge at the reported coordinates.

Type 5 Ground unit, location, time, CAS FM AIR STRIKE list of casualties

Condition: Ground unit reports casualty-producing ordnance impact.

Explanation: Ground unit has incurred the listed personnel and equipment losses from an air strike.

Type 60 Air unit, location, time UNABLE TO CHANGE MISSION - LOW FUEL

Condition:	A change air mission command was issued,but the fuel remaining in the air unit is not sufficient for the unit to carry out the specified mission.
Explanation:	The mission is not changed.
Type 60	Air unit, location,time AIR MISSION IGNORED NO ORDNANCE AVAILABLE FOR TARGET
Condition:	A strike is desired but no ordnance or the wrong type of ordnance is specified for the mission.
Explanation:	Air mission cancelled.
Type 60	Air unit, location,time AIR UNIT MISSION CHANGED
Condition:	A change air mission command is accepted.
Explanation:	Self-explanatory.
Type 60	Air unit, location, time AIRBORNE IN elapsed time
Condition:	An air mission command is accepted.
Explanation:	The desired air mission was checked and accepted. The air unit will take off in the amount of time specified.
Type 60	Air unit, location, time AIR MISSION CANCELLED
Condition:	When an air mission is cancelled.
Explanation:	Self-explanatory.
Type 68	Air unit, location, time DETECTED GROUND UNIT ground unit WITH sensing equipment AT location
Condition:	Air unit freshly detected ground unit with the specified sensing equipment.
Explanation:	Self-explanatory.
Type 69	Air unit, location, time DETECTED GROUND UNIT ground unit WITH sensor AT location MSG. DELAYED - RECEIVED AT time
Condition:	Air unit freshly detected ground unit with a type of sensor that requires processing time (like IR-Cameras).
Explanation:	The alert message is not actually delayed. Only the delay time is indicated.
Type 58	Air unit, location time TOOK SPT FIRE FROM ground unit L = no. of planes lost, R = no. of planes remaining.

Condition:	Air unit flies through a hot support fire zone and incurs losses.
Explanation:	Air unit has suffered losses of the indicated number of planes.
<u>Type 60</u>	<u>Air unit, location, time</u> MISSION ABORTED - VISIBILITY TOO LOW
Condition:	Visibility is lower than the poorest meteorological visibility in which aircraft can continue its mission (ROME(IAC, 7)).
Explanation:	Mission aborted.
<u>Type 60</u>	<u>Air unit, location, time</u> AIR UNIT CRASHED - DENSITY ALT. TOO HIGH.
Condition:	The density altitude is too high for the plane to fly in. ROME (IAC,2) specifies the maximum altitude that the type of plane is allowed to fly in.
Explanation:	Self-Explanatory.
<u>Type 60</u>	<u>Air unit, location, time</u> AIR UNIT CRASHED-FUEL STARVATION
Condition:	Air unit has run out of air fuel.
Explanation:	Since the fuel level is checked before the flight takes off, this case happens rarely. Theoretically, it can happen if the weather changes drastically during a flight.
<u>Type 60</u>	<u>Air unit, location, time</u> AIR UNIT - LANDED
Condition:	Air mission completed.
Explanation:	Self-explanatory.
<u>Type 67</u>	<u>Air unit, location, time</u> RECEIVING GROUND FIRE, L = <u>no. of planes lost</u>, R = <u>no. of planes remaining</u>
Condition:	Air unit was fired at and incurred losses.
Explanation:	The indicated no. of planes are lost.
<u>Type 1</u>	<u>Unit, location, time</u> VIOLATED <u>fire control measure type, control measure name</u>
Condition:	This alert applies only to fire control measures which are not to display only, and only to support fire.

Explanation: Violation of fire control measures occurs in two ways:

1) Fire short of fire control measures that are to be fire past.
2) Fire across fire control measures that are not to be fired across.

Since the automatic fire allocation logic checks for these conditions before firing,these alert messages can be issued only when unit is under command and control.

Type 2 Unit, location, time CROSSED control measure type, control measure name

Condition: This alert applies only to control measures which are not fire control measures and not for display only.

Explanation: Alert messages say the unit has crossed the identified control measure.

Type 61 Ground unit 1 AURALLY DETECTED ground unit 2, location, time

Condition: Noise produced by unit 1 is detected by unit 2.

Explanation: Another alert will not be generated until the detected unit moves a user input threshold distance away from the detecting unit - at which point it again becomes eligible for detection.

Type 62 Ground unit 1, VISUALLY DETECTED ground unit 2, location, time

Condition: Ground unit 1 has just visually detected ground unit 2 at the specified location.

Explanation:

1) The detection logic is stochastic in the sense that the probability of visual detection is compared with a randomly drawn uniformly distributed random number before an alert is generated. This means that the existence of line of sight between units does not necessarily mean automatic visual detection.

2) Visual detection is "remembered" and "degraded" every simulation minute when line of sight with detected unit is temporarily lost. Reinitiated line of sight can however upgrade the fraction of visual detection.
3) When the fraction of visual detection is degraded down to an input threshold or when the detected unit moves an input specified distance away from the detecting unit, visual detection is reset to zero. The cycle starts over when line of sight is reinitiated.

<u>Type 63</u> — <u>Ground unit 1</u> DETECTED WITH RADAR <u>ground unit 2</u>, <u>location, time</u>

Condition: Ground unit 1 has just detected ground unit 2 with its radar equipment.

Explanation: Another message cannot be generated until the unit has moved out of the range of the radar equipment and is then redetected.

<u>Type 64</u> — <u>Color, sensor name</u>, UNATTENDED GROUND SENSOR ALARMS AT <u>location</u>, TIME = <u>time</u>

Condition: The identified blue or red unattended ground sensor has just sensed an intruding unit at the indicated location.

Explanation: Another message cannot be generated for the same unit in the same sensor field.

<u>Type 53</u> — <u>Unit</u> DAMAGED RD AT <u>location, percent</u> PERCENT

Condition: Unit has damaged the indicated percent of the road located at the specified location using its support fire weapons.

Explanation: Self-explanatory.

<u>Type 54</u> — <u>Unit</u> DAMAGED BRG AT <u>location, percent</u> PERCENT

Condition: Unit has damaged the bridge located at the specific location the indicated percentage.

Explanation: Self-explanatory.

Type 76 — Ground unit, location, time RECEIVING FIRE FROM: list of opposing units OF TYPE list of fire types AT A RANGE OF closest range. FIRING: list of fire types

Condition: This message is issued basically when a unit starts engaging in a fire engagement as described in one of the following situations:

1) When unit starts firing a new class of weapon
2) When unit starts receiving a new class of weapon fire
3) When unit starts receiving air ordnance. In this situation, no alert message is generated. The counters for casualties are however started.

Explanation: This alert serves the role of the IN-CONTACT-SITREP used by the Army. Unit reports are its location and the classes of fires (SMALL ARMS, MORTARS, AND ARTILLERY) that it is receiving and firing. A count of the number of enemy units of different sizes along with the distance from the closest one are also reported if the unit is under fire.

A set of counters is initialized at the time when one of these alerts is issued. The counters are used for generating threshold casualty reports and end of fire casualty summaries.

Type 78 — Ground unit, location, time FIRING STARTED time AND ENDED (NOT ENDED) list of casualties

Condition 1: Reporting unit is under fire and equipment and/or personnel casualties since last report have reached a threshold.

Explanation: Reporting unit reports time of the start of firing engagement and indicates that firing has not ended to distinguish from an end-of-engagement casualty summary. A list of casualties incurred since the last threshold casualty report is included in the alert.

Personnel and equipment casualties are accumulated when a firing engagement starts and compard with a threshold each simulation minute. The personnel casualty threshold is specified in fraction of unit personnel killed since the engagement started or since last threshold casualty report. Equipment casualty thresholds are specified in number of pieces lost before a threshold casualty report will be generated.

<u>Type 78</u> — <u>Ground unit, location, time</u> FIRING STARTED <u>time</u> AND ENDED <u>list of casualties</u>

Condition 2: Unit received at least one class of fire last minute and receives none this minute.

Explanation: Reporting unit reports the total personnel and equipment casualties incurred during firing period identified by time 1 and time 2. All casualty counters for the reporting unit are reset after the message and are not set until the next time engagement.

<u>Type 120</u> — END OF SIMULATION

Condition: Simulation has reached its end.

Explanation: Self-explanatory.

<u>Type 82</u> — <u>Unit, location, time, gasoline or diesel</u> LOW = BASIC LD <u>no. of gallons</u>, CURRENT LD. <u>no. of gallons</u>

Condition:

1) Unit must have had fuel and still has equipment that requires fuel.
2) Unit's fuel supply must be below an input specified percentage of the basic load.
3) Unit must not have a low fuel alert of the particular type outstanding since it was last resupplied.

Explanation: Unit will not issue another low fuel alert until it is resupplied with anything.

<u>Type 66</u> — <u>Unit,location, time, was speed 1 KPH mode 1</u> IS <u>speed 2</u> KPH <u>mode 2</u>

Condition: Whenever unit changes from mounted to dismounted or vice versa OR when the change in rate of movement is sufficiently large (> DELMNT) and is bigger than DELMOVE * units last minute movement rate.

Explanation: The unit has either changed its mode of travel or its movement rate has changed significantly.

<u>Type 25</u> — <u>Unit, location,time</u> HALTED BY <u>obstacle type</u> FOR <u>delay time</u> MIN

Condition: Unit has encountered an obstacle.

Explanation: Unit is stopped by the identified obstacle and will take the indicated amount of time to cross the obstacle.

<u>Type 57</u> — <u>Unit, location, time</u> ENCOUNTERED MINEFIELD <u>minefield name</u> LOST <u>personnel and equipment lost</u>

Condition: Reporting unit has run into a minefield and has incurred some injury.

Explanation: The kind of casualty incurred basically depends on the mode of travel.

<u>Type 81</u> — <u>Unit, location, time</u> AMMUNITION LOW

Condition: Unit ammunition level of one or more type is less than AMOREV* origional amount and unit still has weapons that may require such ammunition type(s).

Explanation: Unit will not issue another low ammunition alert until it is resupplied.

<u>Type 53</u> — Road damage resulted from air delivered weapons. Please see DMGEVAL.

<u>Type 54</u> — Bridge damage resulted from air delivered weapons. Please see DMGEVAL.

<u>Type 62</u> — Visual detection. Please see DALERT.

Explanation: When two units are so close to each other that their areas overlap, visual detection is forced.

Type 56	Unit, location, time is REDCON indicator DUE TO personnel or equipment.
Condition:	When the READINESS CONDITION index of a unit changes.
Explanation:	READINESS CONDITION index is calculated according to rules outlined in Appendix D of the Programming Report.
Type 91	Unit 1, location, time, amount, material name FM donor unit
Condition:	Unit 1 has received the indicated amount of the specified material from the identified donor unit.
Explanation:	Self-explanatory.
Condition:	Unit 1, has just given the indicated amount of the specified resource to the recipient unit.
Type 92	Unit 1, location, time, amount given, material TO recipient unit.
Explanation:	Self-explanatory.
Type 119	Time 1 RESTART TO TIME time 2
Condition:	At time 1, the simulation is reset to time 2.
Explanation:	Self-explanatory.
Type 119	Time 1 REINITIALIZE TO SCENARIO scenario name
Condition:	At time 1 the simulation is reinitialized to the identified scenario.
Explanation:	Self-explanatory.
Type 72	Unit, location, time FIRING weapon name AT location
Condition:	Unit fires the identified support fire weapon at the indicated location.
Type 83	Unit, location, time C AND C HQ DESTROYED, COMM LOST FOR no. MIN
Condition:	The personnel level or the vehicle equipment level of the identified command post has been destroyed down to the level that communication is temporarily lost.
Explanation:	The thresholds can be NAMELISTED. During the period when communication is lost, no C & C will be accepted and no alert messages will be issued.
Type 84	Unit, location, time COMMUNICATIONS RESTORED
Condition:	Period for loss of communication with the command post has elapsed.

Type 56	Unit, location, time is REDCON indicator DUE TO personnel or equipment.
Condition:	When the READINESS CONDITION index of a unit changes.
Explanation:	READINESS CONDITION index is calculated according to rules outlined in Appendix D of the Programming Report.
Type 91	Unit 1, location, time, amount, material name FM donor unit
Condition:	Unit 1 has received the indicated amount of the specified material from the identified donor unit.
Explanation:	Self-explanatory.
Condition:	Unit 1, has just given the indicated amount of the specified resource to the recipient unit.
Type 92	Unit 1, location, time, amount given, material TO recipient unit.
Explanation:	Self-explanatory.
Type 119	Time 1 RESTART TO TIME time 2
Condition:	At time 1, the simulation is reset to time 2.
Explanation:	Self-explanatory.
Type 119	Time 1 REINITIALIZE TO SCENARIO scenario name
Condition:	At time 1 the simulation is reinitialized to the identified scenario.
Explanation:	Self-explanatory.
Type 72	Unit, location, time FIRING weapon name AT location
Condition:	Unit fires the identified support fire weapon at the indicated location.
Type 83	Unit, location, time C AND C HQ DESTROYED, COMM LOST FOR no. MIN
Condition:	The personnel level or the vehicle equipment level of the identified command post has been destroyed down to the level that communication is temporarily lost.
Explanation:	The thresholds can be NAMELISTED. During the period when communication is lost, no C & C will be accepted and no alert messages will be issued.
Type 84	Unit, location, time COMMUNICATIONS RESTORED
Condition:	Period for loss of communication with the command post has elapsed.

Explanation:	The way the logic is set up now, a given command post will only lose communication once.
Type 36	WEATHER REPT AT time - TEMPERATURE = temperature, HUMIDITY = no. %, VIS = no. KM.
Condition:	The weather has changed.
Explanation:	Temperature is given in degrees Farenheit. Visibility is given in kilometers.

c) Term References:

The following subroutines generate alert messages of the type indicated.

ADFALERT: 59,70	FIRAGEN: 76,78
ADW: 71	FORMAIN: 120
ADWALRT: 5	LOWALRT: 82,66,25,57,81
AIREVENT: 60	OTHRDMG: 53,54
AIRGRND: 68,69	OVERUN: 62
AIRHOTZN: 58	REDCON: 56
AIRMOV: 60	RESUPPLY: 91,92
AIRMOV2: 67	SAVE: 119
CK4XING: 1,2	SETIMP1: 72
DALERT: 61,62,63,64	STEP: 83,84
DMGEVAL: 53,54	WITHRC: 36

3.12 AMMUNITION TYPE

3.12.1 English-language Definition

a) Term Name:

Ammunition Type

b) Term Description:

Each ground equipment mode of operation for which a firing rate is defined must have an associated ammunition type defined. The effect of this ammunition number is two-fold:

1) The unit ammunition level for that ammo type must be sufficient to allow the unit to fire its weapon in the desired mode of operation

2) The ammunition type is a factor in determining which weapon effects function and data is to be used to calculate casualties (see the discussion of the weapons effects table in the next section)

Aircraft, on the other hand, use this array to store takeoff and landing delay times, which are preset at initialization. Air ordnance types use the array for various delivery parameters.

Data in this array is defined at system initialization and never altered.

c) Term References:

Ammunition type is basically an equipment attribute defined for each ground equipment mode of operation. For aircraft and air ordnance, random attributes as defined above are stored in the ammunition arrays. Non-firing ground equipments and air sensors do not use this array. The Ground fire module and Air module are the principal users.

3.12.2 Programmatical Definition

a) Term Name:

IAMTE(IEQ,IMODE)
IAMMONAM(M,IAMMO)
IAMOSIDE(IND)

b) Term Description:

IAMTE(JEQ,IMODE), IAMMONAM(M,IAMMO), and IAMOSIDE(IND) are all data base input variables, with IAMTE(JEQ,IMODE) being defined in the equipment input deck and IAMMONAM(M,IAMMO) and IAMOSIDE(IND) both defined in the ammunition input deck.

IAMTE (IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0), ammunition type used by weapon type IEQ in mode of operation IMODE (0 implies no ammunition used).

For aircraft (IEQCOD(IAC) = -1),

IMODE = 1 takeoff delay time for this aircraft (min)

2 landing delay time for this aircraft (min)

For air ordnance (IEQCOD(IAC) = -2),

IMODE = 1 number of ammunition type for this equipment, if any

2 number of rounds in a standard load of that ammunition type

3 Rate of fire of this equipment (rounds/min)

4 Number of drops per pass (applies to bombs)

5 Distance between drops (meters)

6 Dud probability times 100

7 Kill probability times 100 for bridge type 1

8 Kill probability times 100 for bridge type 2

IAMMONAM(M,IAMMO) A 12 character (M = 1,3) name for each of the possible (K = 1,80) ammunition types.

IAMOSIDE(IND) A byte packed array (IND = 1,20) where each byte contains a code controlling which ammunition types will appear on the menus when the red or blue command and control button is pushed.

0 = Both on red and blue

1 = Red only

2 = Blue only

3 = Neither Red or Blue (Spare)

c) Term References:

IAMTE(IEQ,IMODE) is indexed primarily on equipment number ($1 \le$ IEQ ≤ 80), secondly on mode of operation ($1 \le$ IMODE ≤ 8). For air equipment, the second index is a dummy index. The principal ground fire subroutines using IAMTE are:

AIRCAS
AMOVUL
ORGFIR
SPTALO

The principal air subroutines using this array are:

ADW
AIREVENT
AIRMOV
OTHRDMG

3.13 AMPHIBIOUS OPERATION

3.13.1 English-language Definition

a) Term Name:

Amphibious Operation

b) Term Description:

Amphibious operation refers to the simulation of movement by ground units over water obstacles. This involves delaying the unit for a period time equivalent to the time required for a unit to prepare and initiate an amphibious landing operation. Currently, the delay time is a user defined input. Once the delay time has elapsed, the unit can resume movement again. Note that if the unit is able to construct a bridge in a shorter time period, it will do so, rather than suffer the delay associated with an amphibious operation.

c) Term References:

Amphibious operation is referenced only by the movement function in the model. Specifically when a unit is halted by a water obstacle, it is given the choice of enduring the delay times associated with building a bridge or conducting an amphibious landing operation. The course of action is determined by the shortest delay time. In either case, the delay time is used to simulate the overall action of crossing over a water obstacle.

3.13.2 Programmatical Definition

a) Term Name:

AMPHBUS

b) Term Description:

AMPHBUS is a data base input variable defined in the Namelist input deck. It is the delay time used to simulate the actions of an amphibious landing operation. AMPHBUS must be a nonnegative floating point number in units of hours. AMPHBUS must be pre-defined by the user, and is input by NAMELIST.

c) Term References:

AMPHBUS is referenced only by subroutine ENGRSPT.

3.14 AREA OCCUPIED

3.14.1 English-language Definition

a) Term Name:

Area Occupied

b) Term Description:

The area that a unit occupies is calculated in the mode to be the product of a unit's prescribed frontage (width) and depth. It is assumed that personnel and equipment are uniformly distributed within this area for purposes of computing weapon effects. Unit frontage and depth become important as separate factors when the unit enters an engagement (see the discussion of the engagement module in Section 5). A graphics display button allows the selection of area occupied by all units.

c) Term References:

Area occupied is a unit attribute represented by unit width and unit depth. It is input at initialization time and can be changed via the "unit location" menu.

3.14.2 Programmatical Definition

a) Term Name:

IUWID(IU),IUDEP(IU)

b) Term Description:

Unit area occupied is an important consideration in the calculation of weapons effectiveness for area fire ground weapons and for air-ordnance delivery as described in detail in Section 5 under the Ground Fire Module and the Air Module. For a given target unit, if all other factors are kept constant, as the area occupied increases, the casulties for the unit decrease, since there are fewer people per unit of area affected by the weapon lethality. Variables IUWID(IU) and IUDEP(IU) are defined as follows:

IUDEP (IU)	Depth of unit IU in meters
IUWID (IU)	Width of unit IU in meters

IUWID and IUDEP are both data base input variables and interactively generated variables. As data base input variables, they are defined in the unit input deck, and, as interactively generated variables, they are created from the unit location menu.

c) Term References:

Variables IUWID and IUDEP are indexed by unit number, where ($1 \leq IU \leq 100$). The arrays are used primarily by the following subroutines:

ADW	for air ordnance lethality
EFFNS	for ground weapon lethality
NEWENG LATDST	for the formation of new engagements

3.15 AREA TARGET

3.15.1 English-language Definition

a) Term Name:

Area Target

b) Term Description:

When an air strike mission is created, the controller must select a target type from the menu. If "area target" is selected, the menu processor passes the designated point to the math model to use as the aim point.

When the ordnance is delivered at the point, the actual point of impact is computed from delivery error data, and a lethal area is built around this impact point for each target element (personnel vulnerability class and ground equipment) in any unit affected by the impact, possibly including a friendly. The fraction of the unit inside the lethal area for target element is used to proportionately reduce the number of those target elements in the unit.

c) Term References:

Area target is a specific designation for an air unit on a strike mission and is set at the time the air unit is created, and used when ordnance is delivered.

3.15.2 Programmatical Definition

a) Term Name:

NTGTYP(IAU) = 201

b) Term Description:

The array NTGTYP(IAU), defined as

NTGTYP (IAU) — Type of target for air unit IAU; = ground unit number or code for road, bridge, or area target.

is an interactively generated variable created from the air menu and is set to 201 as an indicator to subroutine ADW to process the target designation as an area target.

c) Term References:

The index to NTGTYP(IAU) is the air unit number, where ($1 \leq IAU \leq 10$). Subroutine AIREVENT sets NTGTYP(IAU) to 201, so that subroutine ADW delivers the ordnance against a general area target.

3.16 BACKGROUND SOFTWARE

3.16.1 English-language Definition

a) Term Name:

Background Software

b) Term Description:

The CATTS math model is a large, detailed, complex digital computer simulation of the tactical battlefield environment. It is a time-step model, with timesteps of one minute (except for air units, which have quarter-minute steps). It calculates for each minute of battle the detections, engagements, fires, casualties, movement, and environmental effects for up to ninety-nine units. The baseline scenarios have units which vary in strength from a squad to a battalion, with the normal level of platoon for friendly units and company for aggressor units.

Because the model is not interrupt-driven, it runs as a background program, and is often referred to as the "background software." This means that the model is executed at a lower priority than any program residing in the foreground area, which can be referred to as "foreground software". All interrupt driven software is in the foreground. When no requests for processing of a foreground program are queued, the background software is executed.

The model residing in the background is functionally divided into ten modules, each with a specific function. Each module is described in detail in Section 5 of this document.

c) Term References:

(None)

3.16.2 Programmatical Definition

(Not applicable)

3.17 BANDS

3.17.1 English language Definition

a) Term Name:

Bands

b) Term Description:

Fire support bands are used to restrict the fire of fire support weapons(code = 3) to targets within specified, prioritized regions. The bands define the regions into which a support fire weapon should allocate its general support fire. Sets of fire-support target bands are used with modes 6 and 8, (general support modes against unengaged units.) The appropriate set to use with a particular weapon in a specific unit depends on the unit's operational state and is specified by code W of the proper fire support table entry. Associated with each set of bands is a set of band priorities, and band allocation factors to be used either to weight the targets in each band, or to allocate the specified fraction of weapons to the band. For instance, if a certain number of weapons have been allocated to mode 6 and mode 6 targets have been partitioned into fire-support target bands, and code X of the Fire Support Table is 1, then these weapons will be allocated to the various bands in the order of the given band priorities, and the fraction allocated will be the value of the band allocation factor or the fraction of the weapons remaining to be allocated, whichever is less. The sum of the band allocation values for a set of bands should ordinarily total more than 1.0. If code X is 2, band priorities are not used, and the values of the band allocation factors become a set of band weighting factors. These factors are combined with all other weights assigned the mode 6 targets, and fire from the available weapons is then allocated to these targets strictly on the basis of target weig ht.

c) Term References:

Bands are model entities described by the attributes listed in the Programmatical description below:

3.17.2 Programmatical Definition

a) Term Name:

BANDAL(IBDS,IBD)
BDFRAC(IBD,J)
IBANDP(IBS,IBD)
IBANDX(IBDS,IBD)

b) Term Description:

The bands are model entities consisting of the following attributes, with definitions of each:

BANDAL (IBDS,IBD)	For each band set, allocation factor to be used with each band. May be either the fraction of available firepower to be used on the band or the relative weighting factor to be used in assigning firepower to targets in the band. Indeces are same as for IBANDP.
BDFRAC (IBD,J)	Factor of each band by which to multiply a given target allocation value in order to give fraction of weapons to allocate to the target.
IBANDP (IBDS,IBD)	For each band set, the allocation priority assigned to each band. A zero value of IBANDP implies that band is not used in allocation. (IMDS=BAND set, IBD = BAND number for which priority IBANDP (IBDS, IBD is given (1 < = IBDS < = 9, 1 < = IBD < = 4).
IBANDX (IBDS,IBD)	Three X-coordinates (IBD)--defining target bands on battlefield (for general support allocation, modes 6 and 8). The bands or regions defined (extending to X =- infinity and X=infinity on each end) are subject to different allocation factors with regard to targets lying within them. Nine of such band (1< = IBDS < = 9) sets may be defined with the three X-coordinates necessary to describe the four bands.

BANDSUM (I,J) For the Ith band of a band set ($1 \leq I \leq 4$), the total target allocation value in a band, where J=1 represents mode 6 fractions, J=2 represents mode 8 fractions.

These variables are data base input variables that are defined in the fire support input deck.

Nine sets of bands are currently defined by data base input, and not changed during model execution. These bands only affect the automatic support fire allocation, and have no effect on commanded support fire.

c) Term References:

Arrays BANDAL and IBANDP are dimensioned 9 x 4 for 9 possible band sets, 4 bands in each. IBANDX is dimensioned 9 x 3 for 9 possible band sets, 3 vertical (x) boundaries for the 4 bands. BDFRAC(I,J) is dimensioned 4 x 2, where I ranges over the 4 bands, J=1 represents mode 6 fractions, J=2 represents mode 8 fractions; likewise for BNDSUM.

Arrays BANDAL, BDFRAC, and IBANDP are used by SPTALO (BDFRAC is also used by GENFIR). Array IBANDX is used by BDTGTS and GENFIR.

3.18 BASIC LOAD

3.18.1 English-language Definition

a) Term Name:

Basic Load

b) Term Description:

The basic load for personnel, equipment, gas, and diesel fuel for ground units is input at initialization time from the data base and not changed unless the unit is resupplied. The basic load for personnel, equipment and avaiation gas for air units is set at the time an air mission is created.

c) Term References:

The arrays containing basic load data are unit attributes and, as such, some referenced by unit number.

3.18.2 Programmatical Definition

a) Term Name:

PERSI(I),EQINIT(I,J),BLAVGAS(I),BLDIES(I),BLGAS(I)
for equipment, gas, diesel fuel, and aviation fuel, respectively.

b) Term Description:

PERSI (IU)	Initial number of personnel in Unit I.
EQINIT (IU,J)	Initial amount of Jth equipment type in Unit I.
BLAVGAS (IU)	Basic load of aviation fuel for Unit I.
BLDIES (IU)	Basic load (initial amount) of diesel fuel for unit I (1-99) in gallons
BLGAS (IU)	Basic load (initial amount) of gas for unit I(1-99) in gallons

EQINIT(I,J) is a data base input variable defined in the unit input deck. BLAVGAS(I) is a model generated variable created in subroutine AIREVENT, while BLDIES(I) and BLGAS(I) are model generated variables created in subroutine INPUT, and PERSI(I) is created from the data input as MENI in subroutine UNINP.

c) Term References:

All basic load variables are unit attributes and, therefore, their unit index, IU, ranges from ($1 \leq IU \leq 100$). The second index for equipment basic load (EQINIT) ranges from ($1 \leq J \leq 14$) representing up to 14 different equipments that a unit can have. Subroutines using the basic load variables include the following:

AIREVENT	LOWALRT
CHGCRT	REDCON
CRDLIC	RESUPPLY
EQUPLINE	STATREP
FORRAM	STEP
FUELLINE	SUPRES
INPUT	UNINP

3.19 BATTALION

3.19.1 English-language Definition

a) Term Name:

Battalion

b) Term Description:

A battalion is designated in the model by setting the size value for a specific unit to 5. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for battalion is "11". The term battalion is also associated with the command and control panel button selections. When battalion is selected, all unit sizes having a designated value greater than or equal to battalion are displayed on the appropriate menu.

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference. Those units designated a size of 5 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 5 used to represent battalion.

3.19.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is both a data base input variable and an interactively generated variable. As a data base input variable, it is defined in the unit input deck for units 1 - 100, the operational group input deck for units 101 - 120, and the higher and adjacent unit input deck for units 121 - 150. As an interactively generated variable, it is created from the tasking organization menu for units 101 - 120.

A battalion is designated for Unit I in the model by setting array ISIZE(I) to a value of 5 as part of the data base inputs. This value is never altered for units. A battalion is designated for operational group (I-100) by setting array ISIZE(I), ($101 \leq I \leq 120$), equal to 5 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 5 for ISIZE(I), where I is in the range ($121 \leq I \leq 150$), represents the designation of battalion for red and blue adjacent units represented by those unit numbers.

The term battalion is also associated with the command and control panel button selections. When a value greater than or equal to 5 is designated in array ISIZE for a given unit, the unit is displayed when the battalion button is selected.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVFOG	UNINP
NEWFWDUN	USEFUEL

AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

3 OF 8
AD
A038796

3 OF 8

AD
A038796

3.20 BREACHING A MINEFIELD

3.20.1 English-language Definition

a) Term Name:

Breaching a Minefield

b) Term Description:

When a unit encounters a minefield, the unit will suffer personnel casualties and/or damaged equipment and will be detained for a period of time. At the end of the delay period, the unit will breach the obstacle by finding the shortest path across the minefield. Since minefields are modeled as rectangles, the shortest path across is usually the path normal to the side of the rectangle containing the point of obstruction. However, the normal path is not necessarily the shortest path. When breaching a minefield in the CATTS model, the shortest path of travel across the minefield is established and taken after the unit has suffered casualties and delay.

c) Term References:

Breaching a minefield is referenced by the movement function of the model. In particular it refers to the X-Y coordinates of the endpoints defining the shortest path to be traversed through the minefield.

3.20.2 Programmatical Definition

a) Term Name:

Breaching a minefield is represented by a series of calculations that determine the shortest route through the minefield. There is no programmatical term name related to this.

b) Term Description:

Breaching a minefield consists of computing values for the following FORTRAN variables:

ENTRYX,ENTRYY = the X and Y coordinates of the point where the unit enters the minefield.

EXITX,EXITY = the X and Y coordinates of the point where the unit exits the minefield.

These values are used in further calculations which yield the final XY coordinates of the endpoints defining the shortest line segment across the minefield. In particular, the XY coordinates given by EXITX,EXITY are used to compute the destination point for the unit to follow in order to travel across the minefield on the shortest route. The computed XY coordinates of the destination point are transformed so that it can be halfword packed into the Ith element of the array IBEYOND where I is the unit number traversing through the minefield:

IBEYOND(I) = A packed array (half words) containing the following data for the Ith ($1 \leq I \leq 100$) unit:

First half word: X-coordinate (divided by four) of destination point when traveling across an area obstacle.

Second half word: Y-coordinate (divided by four) of destination point when traveling across an area obstacle.

c) Term References:

The above mentioned FORTRAN variables are used in one or more of the subroutines listed below:

BRCHPATH
FINDBRDG
OBSDELAY
OBSUPDAT
OBSWIDTH

3.21 BREAK RANGE

3.21.1 English-language Definition

a) Term Name:

Break Range (also see table definition for this term in Section 4.)

b) Term Description:

Break ranges are a pair of values used to determine which of three made distribution vectors a unit should use to allocate people and equipment over the various modes of operation. The ranges are in units of meters. A calculation is performed for a particular unit to determine either the range to the closest enemy unit or the range to the enemy FFBA, whichever is smaller. The mode selection code table is searched to find the appropriate set of break ranges and corresponding mode distribution vectors. If the unit range calculation is less than the first break range, the first mode distribution vector is used, etc.

c) Term References:

The appropriate set of break ranges and mode distribution vectors is determined by matching a code word in the mode selection code table with

1) the firing unit's operational state
2) the target unit's operational state
3) the target unit type
4) the equipment type being fired

3.21.2 Programmatical Definition

a) Term Name:

MURTAB(I,J)

b) Term Description:

MURTAB(I,J) is a data base input variable that is defined in the mode input deck. It is a two dimensional table containing two break ranges

and three mode distribution vector numbers for each row I. The table is used by any unit preparing to fire a weapon at one or more target units. Having determined from the mode selection code table which entry of the Break Range table is appropriate, the range between the firing unit and the nearest target unit is compared with each of the break ranges. If the target range is less than the first break range, the first mode distribution vector number (indexed by (I,3) is used if the target range is between the two break ranges, the second mode distribution vector number is used; if the target range is greater than the second break range, the third mode distribution vector number is used. This selection of mode distribution vector is extremely important in the ground fire and ground movement modules. The units for break range are meters. See Section 4 (tables) for a detailed explanation of the use of this table.

c) Term References:

MURTAB(I,J) is a two-dimensional table. Index I represents the row of the table to be used for the unit being processed, as determined by the MUSLCT table of code words. I ranges between ($1 \leq I \leq 44$). Index J ranges between ($1 \leq J \leq 5$), where the first two values of J reference the two break ranges, and the last three the associated mode distribution vector numbers. The following subroutines use MURTAB:

MUINP
SCHMU

3.22 BRIDGE

3.22.1 English-language Definition

a) Term Name:

Bridge

b) Term Description:

Bridges in the CATTS mathematical model are defined by the following attributes:

- The x-y coordinates of the endpoints of the bridge.
- The type of bridge (metal or pontoon).
- The water obstacle that the bridge crosses over.
- The amount of damage, if any, against the bridge.

Currently, bridges can be pre-defined via input, or automatically during the simulation when it is feasible for a unit to construct one to cross over a water obstacle. A maximum of sixteen bridges can exist simultaneously.

c) Term References:

Bridge number is used to reference the data listed above under term description. Bridges are used primarily for two major functions in the model. The movement function determines if, when a water obstacle is encountered, an undamaged bridge is available in the vicinity to facilitate traversing it. The firing function, both ground and air, causes damage to bridges affected by air and ground attack.

3.22.2 Programmatical Definition

a) Term Name:

Bridge is an entity represented in the model by the attributes listed under "Term Description"; it is not represented by any single variable.

b) Term Description:

The description of a bridge consists of the following program variables:

IBRIDGEX(IBR,M)	=	First (m=1) and second (M=2) X coordinate of bridge IBR
IBRIDGEY(IBR,M)	=	First (M=1) and second (M=2) Y coordinate of bridge IBR
IBRGTYPE(IBR)	=	Type of bridge 1 = metal or stone 2 = wood or pontoon
BRDDMGE(IBR)	=	Fraction of bridge IBR damage
IBRGSPAN(IBR)	=	Number of obstacle which bridge IBR crosses

BRDDMGE(IBR) is a model generated variable that is used in the Air and Ground Fire Modules. All other variables are both data base input variables that are defined in the bridge deck and model generated variables that are used in the Ground Movement Module.

c) Term References:

The above listed variables are indexed on bridge number, represented by IBR in the nomenclature list ($1 \leq IBR \leq 16$). These variables are used in one or more of the subroutines listed below.

AIRHOTZN	POINTGT
BRRDCHCK	RDSON
BULDBRDG	ROADINP
DMGEVAL	SETIMP2
FINDBRDG	SPTALO
OTHRDMB	

3.23 BRIDGE TYPE

3.23.1 English-language Definition

a) Term Name:

Bridge Type

b) Term Description:

Two types of bridges are represented by the CATTS model. Type one bridges are sturdy metal or stone bridges which are less susceptible to damage from air or ground ordnance. Type two bridges are wood or pontoon bridges which are more vulnerable to attack. Units which constrict bridges to cross over water obstacles always build bridges of type two.

c) Term References:

Bridge type is an attribute of bridge, and is referenced by the bridge number (1 through 16). Bridge type is used mainly by the firing function to distinguish the width of the bridge. Bridge width is an important factor in determining damage inflicted upon it by air and ground ordnance. Bridge type is specified by the movement function when a unit builds a bridge to cross over a water obstacle. Bridges so constructed must be of type two.

3.23.2 Programmatical Definition

a) Term Name:

IBRGTYPE(16)

b) Term Description:

IBRGTYPE(M) is a data base input variable that is defined in the bridge input deck. It is stored in an integer array of dimension sixteen (sixteen bridges may exist concurrently). IBRGTYPE must contain the integer value 1 or 2 specifying bridge type. The value can be pre-defined by input or computed when a bridge is constructed during the simulation

c) Term References:

IBRGTYPE is indexed on bridge number, represented by IBR in the nomenclature list ($1 \leq IBR \leq 16$) IBRGTYPE is used in the following subroutines:

BULDBRDG
DMGEVAL
OTHRDMB
RDSON
ROADINP
IBRGTYPE, itself is an index which references the array IBRGWDTH, this array stores the width of each type of bridge IBRGWDTH, is used in the following subroutines:
DIDITHIT
RDSON
ROADINP

3.24 BRIGADE

3.24.1 English-language Definition

a) Term Name:

Brigade

b) Term Description:

A brigade is designated in the model by setting the size value for a specific unit to 7. This causes the approprite size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for brigade is "X".

c) Term References:

The foreground graphic display programs refers to the size of each unit number being the direct reference. Those units designated a size of 7 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 7 used to represent brigade.

3.24.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit input deck for units 1 - 100, the operational group input deck for units 101 - 120, and the higher and adjacent unit input deck for units 121 - 150.

A brigade is designated for Unit I in the model by setting array ISIZE(I) to a value of 7 as part of the data base inputs. This value is never altered for units. A brigade is designated for operational group (I-100) by setting array ISIZE(I), ($101 \leq I \leq 120$), equap to 7 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 7 for ISIZE(I), where I is in the range ($121 \leq I \leq 150$), represents the designation of brigade for red and blue adjacent units represented by those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP
FIRAGEN
FIXLIST
FORRAM
LEAVEOG
NEWFWDUN
OGINP
OGLOC
STEP
TASKORG
UNINP
USEFUEL

3.25 CASUALTY AND DAMAGE ASSESSMENT

3.25.1 English-language Definition

a) Term Name:

Casualty and Damage Assessment

b) Term Description:

Casualty and damage assessment is performed 1) in the Weapon Effects submodule of the Ground Fire module for ground-ground combat, 2) in the air-delivered weapons portion of the Air module for air-ground combat, and 3) in the air defense portion of the Air module for ground-air combat. Section 5 comtains a detailed discussion of all casualty-producing equations in the appropriate modules. Casualty and damage assessment can be divided into two basic categories. First, personnel casualties are computed for each personnel vulnerability class. As they are incurred, the personnel vulnerability class levels of the target are reduced so that firing of other weapons by the opposition does not account for additional kills of the same people. Personnel vulnerability class levels, therefore, represent the number of target elements available to be fired at in a unit, and as personnel casualties are incurred, these levels of target elements are reduced. Total personnel casualties in a unit, however, are accumulated over the entire time step of firing, and reduced from the unit's total personnel level after all firing has been completed. This reduction in personnel affects the unit's combat power in the subsequent time step by permanently reducing the capability of the unit to man equipment.

The second effect of fire is on equipment damage. Equipment damage assessment is computed and accumulated on each equipment in a target unit. For each weapon firing against an opposing unit, damage assessment is performed against all equipments for which the firing weapon has an effect defined. In computing the damage assessment for an equipment type, the number of pieces existing at the beginning of the minute is used by all weapons firing at the target unit. As equipment damage is incurred, it is accumulated, and at the end of

the time step the unit's equipment level is reduced. This reduction effects the unit's effectiveness in the subsequent minute.

An additional casualty accumulation is kept for the entire length of the exercise. This table accumulates casualties for personnel and each type of ground equipment for Blue as is caused by each Red equipment, and vice versa. There are two additional columns for losses resulting from (1) air-delivered ordnance and (2) other occurrences (maintenance attrition, minefield encounters, etc.) for both Blue and Red. Air unit losses consisting of aircraft, ordnance, sensors, and personnel are reduced from air units immediately.

c) Term References:

The tables accumulating total casualties within a time step are unit attributes. The table of total casualties per equipment for the entire run is an equipment attribute. All three tables are updated whenever casualties are computed. This occurs primarily when weapons effects are are calculated for ground fire weapons, and when air-delivered weapons effects are calculated for air ordnance.

3.25.2 Programmatical Definition

a) Term Name:

DPERS(IU)
DNU(IU,J)
STATS(I,J,K)

b) Term Description:

The following tables are used to accumulate casualty and damage assessment data. Tables DNU and DPERS are accumulated each minute and STATS over the entire exercise.

DNU (IU,J)	Casualties for equipment J in unit IU during current time step.
DPERS (IU)	Personnel casualties for unit IU during current time step.

STATS (I,J,K) — Equipment and personnel casualty statistics for red killing blue (k=1), and blue killing red (K=2) equipments. I=1,40 corresponds to NDXSTATS (I,K) J=1,41 corresponds to NDXSTATS (J,3-K), where J=41 is used for personnel casualties.

See the detailed explanation of the STATS table in the next section.

DPERS, DNU, and STATS are model generated variables. DPERS is used in subroutines ADW, INIT, OBSDELAY, WPNEFF, and WPNFIR. DNU and STATS are used in subroutines ADW, INIT, OBSDELAY, REDIST, STEP, and WPNFIR.

c) Term References:

DNU and DPERS are indexed by unit number ($1 \leq IU \leq 100$). DNU is also indexed by equipment position within the unit ($1 \leq J \leq 14$). The indexing scheme for the STATS array is described in the next section, when STATS is discussed in detail. It suffices to say that the first two indices are related to the equipments on the opposing sides, one for Red, the other for Blue. The third index indicates which is which. Index I ranges from ($1 \leq I \leq 40$) for a maximum of 38 ground equipments for each side, one row for air losses, one row for other losses. The second index, J, ranges from ($1 \leq J \leq 41$), where the first 38 columns represent the maximum number of ground equipments, J = 41 contains personnel losses, and J = 39,40 are not used. The third index, K, represents color, K=1 representing Blue losses to Red equipments, K=2 representing Red losses to Blue equipments.

The primary subroutines updating DNU,DPERS, and STATS are ADW for casualties from air and WPNFIR for casualties from ground firing. Subroutine STEP uses the accumulations in DNU and DPERS to update unit status. Subroutine CASREP prints a summary report of the accumulated casualties per equipment for the entire run.

3.26 CASUALTY THRESHOLD

3.26.1 English-language Definition

a) Term Name:

Casualty Threshold

b) Term Description:

Interim casualty statistics are accumulated for personnel and each equipment in all units every time step. The accumulation begins the first time step that a unit begins receiving any type of fire, and ends when all fire has ceased on that unit. If within that time the casualty accumulation for personnel or any type of equipment in a unit exceeds predefined thresholds, a casualty alert is issued for the unit reporting all accumulated casualties up to that time, including those not exceeding their threshold, if any. The casualty alert is generated following the completion of all firing activity in a given time step. When all firing ceases on a unit, a casualty alert is generated without checking the thresholds. All accumulated casualties since firing began is reported. The accumulating counters are cleared for subsequent enemy contact.

c) Term References:

The actual threshold values for the various equipment types is an equipment attribute, a different value defined for each equipment type. The personnel threshold is a single value, not varying over vulnerability classes, for example. The interim casualty counters are unit attributes accumulated at the end of each time step after all firing has been completed. The fire alert generation function checks the casualty counters against the thresholds to determine if an alert should be issued for that unit.

3.26.2 Programmatical Definition

a) Term Name:

PCSLR

PCTH

CSLR(J,IU)

IECTH(IEQ)

b) Term Description:

The following variables are used in conjunction with casualty threshold. PCSLR accumulates personnel casualties per unit each time step. PCTH is the actual threshold value input a initialization. CSLR accumulates equipment casualties per unit each time step. IECTH is the actual threshold value input at initialization.

PCSLR (IU)	The IUth halfword is a counter of the accumulated personnel casualties of unit IU since shelling starts or since last temporary casualty report.
PCTH	Personnel casualty threshold. A temporary casualty report is generated for a unit whenever percent lost of personnel reaches PCTH*100 since the last report comparison is made with variables COERC and PCSLR in common FIRALRT.
CSLR (J,IU)	CSLR is a halfword array of 14 by 100. Fourteen halfwords (one per equip that each unit may have) is used for each unit to store the total equipment casualties incurred by unit IU since unit IU began receiving fire.
IECTH (IEQ)	Number of pieces of equipment type IEQ which may be lost by a unit before a casualty alert is generated.

PCSLR, PCTH, CSLR, and IECTH are model generated variables. PCSLR is used in subroutine FIRAGEN. PCTH is used in subroutine FORMAIN (data statement). CSLR is used in subroutines FIRAGEN and RESUPPLY. IECTH is used in subroutine FORMAIN (data statement).

c) Term References:

Variable CSLR(J,IU) is a halfword array indexed by unit number as the second index, where ($1 \leq IU \leq 100$). Since each unit can contain up to 14 equipments, ($1 \leq J \leq 7$), each word containing casualty data for two equipment types. Subroutine FIRAGEN accumulates casualties incurred from the current minute into the accumulated casualty counter, then compare that with the appropriate threshold to determine if an alert should be issued.

3.27 CHANGE OF STATE

3.27.1 English-language Definition

a) Term Name:

Change of State

b) Term Description:

The term change of state is defined as a change in the actions of a unit via a change to its operational state caused by the satisfaction of a specific criterion. For example, the state of a unit may change from "moving to contact" to "halted" when the unit arrives at a designated location or when it comes within range of the enemy's direct-fire weapons. A unit may also change its state as a result of changing its movement code. Conditions for changing a unit's state, or change-of-state criteria, are provided through the input of coded decision rules and data. These rules are called "change of state" selection codes. Each of these rules is checked continually to determine whether it is applicable to the unit in question under the unit's current circumstances. If so, a specified criterion is called from a library of such criteria, together with appropriate data. If the criterion is satisfied, the unit changes to the new state specified. Criteria may involve time, unit location, distance from the enemy, the state of other units (friendly and enemy, force ratios, rates and percentages of loss, level of ammunition, and other factors.

c) Term References:

The change-of-state has the function of automatically changing unit operational states automatically. A thorough discussion of the use of this table is contained in Section 4.

3.27.2 Programmatical Definition

a) Term Name:

IOPTB1(I),IOPTB2(I),NDCVAL,DECVAL(I)

b) Term Description:

The change of state table (described in Section 4) consists of three separate arrays, defined below:

IOPTB1 (I,J) Change-of-state selection table for use in changing operational state of unit. Each entry I is of the from UUVVWXXYYZZ, where

IOPTB1 (I,1) = UUVVW
IOPTB1 (I,2) = XXYYZZ
(1 < = I < = 220)
such that
UU = unit type
VV = operational grouping
number of unit (99 ⇒ any unit not in a grouping)
W = 1 or 2 (red or blue)
XX = movement code of unit (88 ⇒ any nonengaged code (5-16))
YY = present operational state of unit
ZZ = unit number.
0 in any position means equality of look-up

IOPTB2 (I) Table of code words associated with words in change-of state selection table. Each entry I is of the form AABBCCD, where

IOPTB2 (I) = AABBBCCD
(1 < = I < = 220)

such that
AA = number of change-of-state criterion to be used
BBB = line number of data value in DECVAL table to be used with criterion
CC = new operational state of unit if criterion is satisfied
D = 1 if unit leaves its operational grouping when it changes state,
= 0 if it does not.

DECVAL (I) The data value with Ith change-of-state criterion.

The variable NDCVAL indicates the size of these tables in use:

NDCVAL Total number of data values in array DECVAL to be used with the criteria in the change-of-state selection table, IOPTB1.

IOPTB1 is both a data base input variable and a model generated variable. As a data base input variable, it is defined in the change of state input deck, and, as a model generated variable, it is used in subroutine CHGOPN. IOPTB2, DECVAL, and NDCVAL are data base input variables that are defined in the change of state input deck.

Subroutine OPPLAN determines if any circumstance in the battle has caused any unit to be considered for a change of operational state. The sequence of processing is as follows:

Each unit is examined in CHGOPN against a list of code decision rules called "change-of-state selection codes") in Table IOPTB1. If a rule i.e., table entry is found that is applicable to the unit in its current operating situation, a specified criterion, together with appropriate input data, is called by CHGCRT. If the criterion is satisfied coded AA in the definition of IOPTB2 above, the unit changes to the new state specified. If not, the check against the list of decision rules is continued.

Units that change state in this manner are examined immediately in NEWMOV to determine whether their movement codes should be changed. The decision rule list is then rechecked for any further change of state, but a unit is not allowed to change its state by this means more than twice during one time step.

A permanent library of change-of-state criteria is stored in the model in CHGCRT. Criteria in this library represent a wide variety of circumstances under which a unit might change its manner of operation. They involve time, unit location (both absolute and relative to friendly and enemy forces), actions of other units, casualties sustained, force ratios, unit suppression, levels of ammunition and equipment, and many other factors.

Presently, just two change-of-state entries are defined.

c) Term References:

IOPTB1 and IOPTB2 are dimensioned 220 and DECVAL is dimensioned 150, with the number of actual entries contained in NDCVAL. Subroutines CHGOPN AND CHGCRT are the only users of these tables.

3.28 CHECK POINT

3.28.1 English language Definition

a) Term Name:

Check Point

b) Term Description:

An air route consists of from 4 to 9 checkpoints designated at the time an air unit is created via the air menu. Each checkpoint consists of a velocity, an altitude, and an XY location. The first point is assumed to be the takeoff point and the last the landing point (altitude and velocity automatically overridden to zero). The air movement function uses the other points to interpolate interim altitude, velocity, and X,Y position at each quarter minute, since all air unit processing is done on a quarter-minute, rather than full minute interval. A check is made when the air unit is defined to ensure that the altitude and speed specified do not exceed aircraft performance.

c) Term References:

Checkpoint is an entity in the model described by the attributes defined above. These attributes are also air unit attributes, since checkpoints are associated with specific air units.

3.28.2 Programmatical Definition

a) Term Name:

A check point is an entity in the model, represented by the following variables:

IACPV(ICKPT,IAU)
IACPX(ICKPT,IAU)
IACPY(ICKPT,IAU)
IACPZ(ICKPT,IAU)

b) Term Description:

The following program variables together describe a checkpoint:

The following program variables together describe a checkpoint:

IACPV (JCKPT,JAU)
velocity at JCKPT-th check point for air unit JAU

IACPX (JCKPT,JAU)
X-coordinate of JCKPTth check point for air unit JAU.

IACPY (JCKPT,JAU)
Y-coordinate of JCKPTth check point for air unit JAU.

IACPZ (JCKPT,JAU)
Z-coordinate of JCKPTth check point for air unit JAU.

These variables are interactively generated variables that are created in the air menu. They are in common block air routes.

c) Term References:

The arrays IACPV,IACPX,IACPY, and IACPZ are indexed initially by checkpoint number ($1 \leq ICKPT \leq 9$), secondly by air unit number ($1 \leq IAU \leq 10$). Checkpoints are set in AIREVENT, the subroutine which processes the air mission event notice. They are used primarily in subroutine AIRMOV to compute air unit positions within each quarter-minute.

3.29 CLOSE SUPPORT

3.29.1 Englísh-language Definition

a) Term Name:

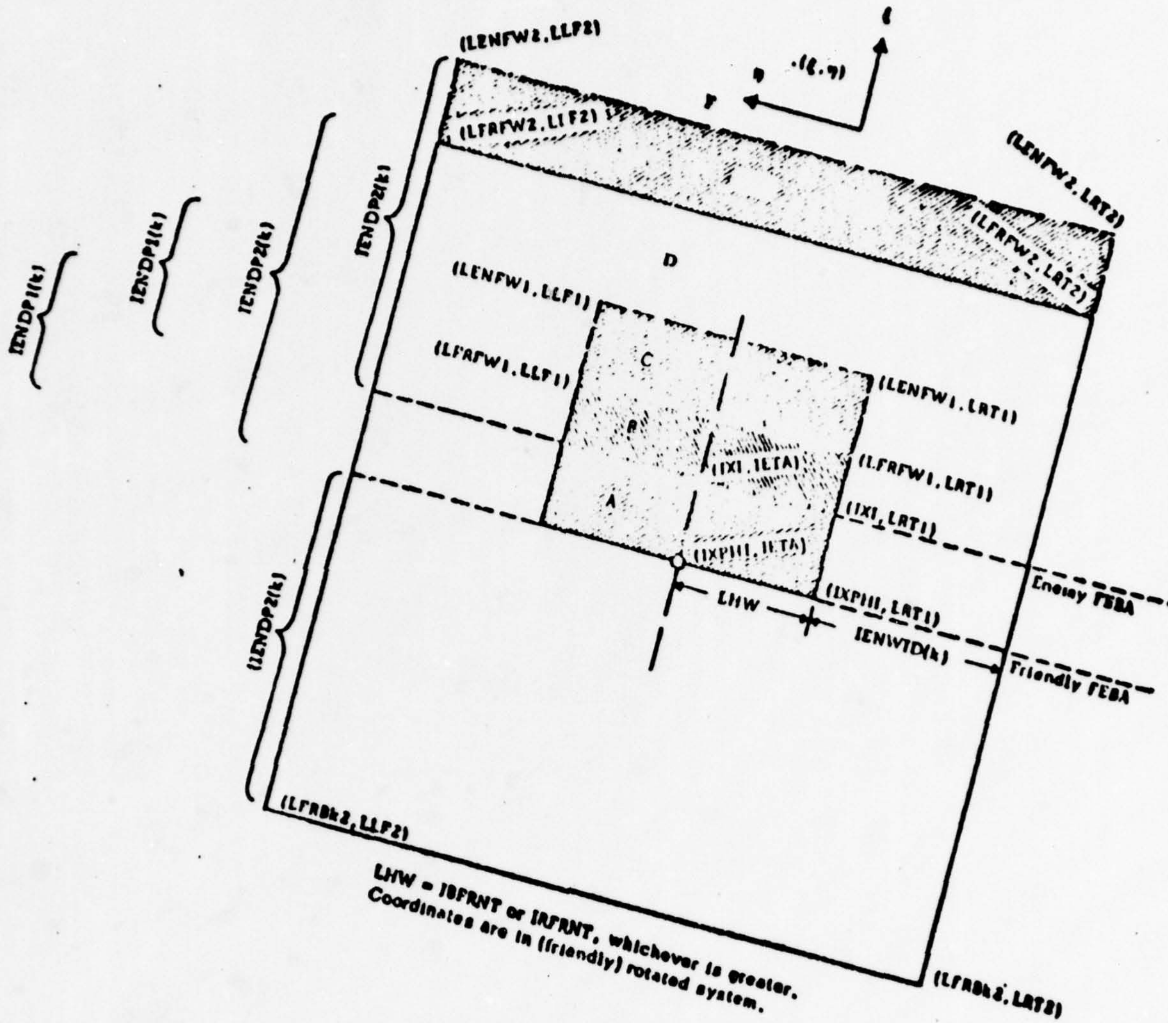
Close Support

b) Term Description:

Two support fire modes of operation are defined as close support (modes 3 and 4). These two modes represent fire at enemy units that pose a direct threat to friendly forces in an engagement. The areas into which close-support fire is directed are defined by rectangular regions in established engagements. Each region includes the enemy FEBA and a number of enemy units, the number being dependent on the depth of fire desired. Fire can be directed at targets with each region or distributed uniformly over the entire region (see code V in the fire support table (IFSTAB) description. Figure 3-1 illustrates the close-support region of an engagement. If such a region is defined relative to the friendly FEBA, it consists of subregions A and B in the figure; if it is defined relative to the enemy FEBA, it consists of subregions B and C in the figure (see code U in the fire support table). The width of a close-support region is the length of the Red or Blue FEBA, whichever is greater. Other dimensions shown on the figure correspond to input values entered as part of the fire support inputs. Any enemy unit lying within one of these regions is classed as a close-support target. Two close-support modes are reserved to permit the use of two different firing rates, mode 3 for normal close support and mode 4 for final protective fires.

c) Term References:

The close support area is used by the automatic suppor- fire allocation function to determine which enemy units are in the forward area of an engagement. The depth of the close support area is input at initialization, whereas the width of the close support area is a function of the engagement.



Note: Nature of target (close support or interdiction) is determined relative both to friendly FEBA (for U = 1 in IFSTAB) and to enemy FEBA (for U = 2).

Figure 3-1. Close Support and Interdiction Target Areas

3.29.2 <u>Programmatical Definition</u>

a) Term Name:

IENDP1(ICOLOR),IBFRNT(I),IRFRNT(I)

b) Term Description:

IENDP1(ICOLOR)	In an engagement, depth of region in which close-support targets may be assigned. This depth is measured either from friendly FEBA or enemy FEBA, depending upon value of fire support weapon code. INCOLOR = 1 Red Firing against Blue INCOLOR = 2 Blue firing against Red.
IBFRNT (I)	Half-frontage of blue force in the Ith engagement.
IRFRNT (I)	Half-frontage of Red force in the Ith engagement.

IENDP1 is a data base input variable that is defined in the fire support deck. IBFRNT and IRFRNT are model generated variables that are used in subroutines DEPLOY, FRNTGS, LATDST, and NEWENG.

c) Term References:

IENDP1(ICOLOR) for ICOLOR = 1,2 are input values defined at initialization time. IBFRNT (IENG) and IRFRNT (IENG) are calculated for engagement number, IENG, where ($1 \leq$ IENG ≤ 12). The following subroutines use one or more of the above variables:

ADJDIR	FRNTGS
CLSPTG	FSPINP
DEPLOY	LATDST
FINDFO	DEWENG
FORRAM	NGARGN

3.30 COLOR

3.30.1 English-language Definition

a) Term Name:

Color

b) Term Description:

Two colors are used by the CATTS math model to distinguish the two armies being simulated. The colors red and blue are used to represent the two armiys respectively. The CATTS math model adops the following code convention when referring to each army by color: the color red is indicated by the integer 1, and the color blue is indicated by the integer 2.

c) Term References:

The referencing of armies by color is used throughout the CATTS math model.

3.30.2 Programmatical Definition

(Not applicable).

a) Term Name:

NUNIT, NRU, NRUP1

b) Term Description:

Variables where the above convention is used are IEQSIDE(IEQ), IAMOSIDE(IAMMO), IOPTB1(I,J), MVCHG1(I,J), and MVCHG3(I,J). The unit colors are determined on the data base by:

NRU - number of red units (red units are from 1 to NRU).

NRUP1 - number of first blue unit (blue units are from NRUP1 to NUNIT).

NUNIT - total number of units (red and blue) must be <=99).

3.31 COMBAT UNIT

3.31.1 English-language Definition

a) Term Name:

Combat Unit

b) Term Description:

The term combat units is used to refer to units selectable under the graphic display button as "Combat". When this button is depressed in conjunction withthe "Area Occupied" button, all units of the selected color having a unit/arm/branch designation of one of the following is to be displayed on the monitor:

INFANTRY
MECH INFANTRY
AIRMOBILE INFANTRY
AIRBORNE INFANTRY
ARMOR
CAVALRY
ARMORED CAVALRY

A unit/arm/branch designation is input for each unit as part of the data base inputs.

c) Term References:

The unit/arm/branch applies to each unit, and as such to referenced by a unit index, thereby being a unit attribute. The term combat is also, therefore, a unit attribute, although the term itself is not directly referenced by unit number.

3.31.2 Programmatical Definition

a) Term Name:

ITPPL(I)

b) Term Description:

The variable ITPPL(I) is a data base input variable that is defined in the unit input deck. It is a unit attribute taking on the following values:

1 = INFANTRY
2 = MECH INFANTRY
3 = AIRMOBILE INFANTRY
4 = AIRBORNE INFANTRY
5 = ARMOR
6 = CAVALRY
7 = ARMORED CAVALRY
8 = ANTI-TANK
9 = ARTILLERY,TOWED
10 = ARTILLERY,SP
11 = AIR DEFENSE
12 = ENGINEER
13 = SIGNAL
14 = MAINTENANCE
15 = MEDICAL
16 = ORDNANCE
17 = QUARTERMASTER

When the value for a given unit lies between 1 and 7, inclusive, the unit is considered combat for purposes of display.

c) Term References:

Although there is no direct reference to combat units by unit number, the index to the array ITPPL represents a unit number. The values that ITPPL can assume are broken into three categories, of which combat is the first. The index of ITPPL(I) ranges over ($1 \leq I \leq 100$), covering all unit numbers. The following background subroutine use ITPPL:

FORRAM
UNINP

3.32 COMBAT SUPPORT UNIT

3.32.1 English-language Definition

a) Term Name:

Combat Support Unit

b) Term Description:

The term combat support units is used to refer to units selectable under the graphic display button as "Combat". When this button is depressed in conjunction with the "Area Occupied" button, all units of the selected color having a unit/arm/branch designation of one of the following is to be displayed on the monitor:

ANTI-TANK
ARTILLERY,TOWED
ARTILLERY,SP
AIR DEFENSE
ENGINEER

A unit/arm/branch designation is input for each unit as part of the data base inputs.

c) Term References:

The unit/arm/branch applies to each unit, and as such to referenced by a unit index, thereby being a unit attribute. The term combat is also, therefore, a unit attribute, although the term itself is not directly referenced by unit number.

3.32.2 Programmatical Definition

a) Term Name:

ITPPL(I)

b) Term Description:

The variable ITPPL(I) is a data base input variable that is defined in the unit input deck. It is a unit attribute taking on the following values:

1 = INFANTRY
2 = MECH INFANTRY
3 = AIRMOBILE INFANTRY
4 = AIRBORNE INFANTRY
5 = ARMOR
6 = CAVALRY
7 = ARMORED CAVALRY
8 = ANTI-TANK
9 = ARTILLERY,TOWED
10 = ARTILLERY,SP
11 = AIR DEFENSE
12 = ENGINEER
13 = SIGNAL
14 = MAINTENANCE
15 = MEDICAL
16 = ORDNANCE
17 = QUARTERMASTER

When the value for a given unit lies between 1 and 7, inclusive, the unit is considered combat for purposes of display.

c) Term References:

Although there is no direct reference to combat units by unit number, the index to the array ITPPL represents a unit number. The values that ITPPL can assume are broken into three categories, of which combat is the first. The index of ITPPL(I) ranges over ($1 \leq I \leq 100$), covering all unit numbers. The following background subroutine use ITPPL:

FORRAM
UNINP

3.33 COMBAT SERVICE SUPPORT UNIT

3.33.1 English-language Definition

a) Term Name:

Combat Service Support Unit

b) Term Description:

The term combat service support units is used to refer to units selectable under the graphic display button as "Combat Service Support". When this button is depressed in conjunction with the "Area Occupied" button, all units of the selected color having a unit/arm/branch designation of one of the following is to be displayed on the monitor:

SIGNAL
MAINTENANCE
MEDICAL
ORDNANCE
QUARTERMASTER

A unit/arm/branch designation is input for each unit as part of the data base inputs.

c) Term References:

The unit/arm/branch applies to each unit, and as such is referenced by a unit index, thereby being a unit attribute. The term combat service support is also, therefore, a unit attribute, although the term itself is not directly referenced by unit number.

3.33.2 Programmatical Definition

a) Term Name:

ITPPL(I)

b) Term Description:

The variable ITPPL(I) is a data base input variable that is defined in the unit input deck. It is a unit attribute taking on the following values:

1 = INFANTRY
2 = MECH INFANTRY
3 = AIRMOBILE INFANTRY
4 = AIRBORNE INFANTRY
5 = ARMOR
6 = CAVALRY
7 = ARMORED CAVALRY
8 = ANTI-TANK
9 = ARTILLERY,TOWED
10 = ARTILLERY,SP
11 = AIR DEFENSE
12 = ENGINEER
13 = SIGNAL
14 = MAINTENANCE
15 = MEDICAL
16 = ORDNANCE
17 = QUARTERMASTER

When the value for a given unit lies between 13 and 17, inclusive, the unit is considered combat service support for purposes of display.

c) Term References:

Although there is no direct reference to combat service support by units, the index to the array ITPPL represents a unit number. The values that ITPPL can assume are broken into three categories of which combat service support is the third. Its index of ITPPL(I) ranges over ($1 \leq I \leq 100$), covering all unit numbers. The following background subroutines use ITPPL:

FORRAM
UNINP

3.34 COMMAND AND CONTROL

3.34.1 English-language Definition

a) Term Name:

Command and Control

b) Term Description:

Command and control is used in the CATTS system to command units to respond to controllers' wishes and to control the environment in which units operate. Command and control directives are input to the CATTS math model at the controllers' stations. Each station contains a set of switches which are labelled for various command and control functions. By selecting one of these switches, a controller initiates a command and control sequence. The sequence is completed by selecting by use of a graph pen various options from the command and control menu displayed on the controller's t.v. monitor and selecting the option "DONE" or "REPEAT" when finished. The command and control options available to a controller are the following:

1. Task Organization
2. Control Points
3. Control Lines
4. Control Areas
5. Preplanned Missions
6. Fire Control
7. Maneuver Control
8. Resupply
9. Air Defense
10. Air Missions
11. Weather
12. Unit Relocation
13. Unit Deactivation

The result of the completion of a command and control sequence is stored on disk in the form of an event notice. Once each math model time step the math model reads the disk and acts upon any event notices that it finds.

c) Term References:

Command and control requests are referenced by the math model event processor in the form of event notices.

3.34.2 Programmatical Definition

(Not applicable)

3.35 COMMAND POST

3.35.1 <u>English-language Definition</u>

a) Term Name:

Command Post

b) Term Description:

A command post is designated in the model by setting the size value for a specific unit to be negative. This causes the command post display symbol to appear for that unit at the unit's location when the command post button is selected.

The symbol for command post is 

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a negative size have the command post symbol displayed at the unit's location. Operational groups and adjacent units do not have command post designations assigned to them.

3.35.2 <u>Programmatical Definition</u>

a) Term Name:

ISIZE(IU)

b) Term Description:

ISIZE(IU) is a data base input variable that is defined in the unit input deck for units 1-100, the operational group input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150.

A command post is designated for unit IU in the model by setting array ISIZE(IU) to a negative value as part of the data base inputs. This value is never altered for units.

The negative value for a command post is not used for op groups or adjacent units ($101 \leq IU \leq 150$).

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index I, ranges from I1 $\leq$ I $\leq$ 100) for normal units, (101 $\leq$ I $\leq$ 120) for op groups, and (121 $\leq$ I $\leq$ 150) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.36 COMPANY

3.36.1 English-language Definition

a) Term Name:

Company

b) Term Description:

A company is designated in the model by setting the size value for a specific unit to 4. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for company is "1".

The term company is also associated with the command and control panel button selections. When company is selected, all unit sizes having a designated value equal to company are displayed of the appropriate menu.

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

3.36.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is both a data base input variable and an interactively generated variable. As a data base input variable, it is defined in the unit input deck for units 1-100, the operational group input deck for units 101-120, and the higher and adjacent units input deck. As an interactively generated variable, it is created from the tasking organization menu for units 101-120.

A company is designated for unit I in the model by setting array ISIZE(I) to a value of 4 as part of the data base inputs. This value is never altered for units. A company is designated for operational group (I-100) by setting array ISIZE(I), ($101 \leq I \leq 120$), equal to 4 to designate the appropriate size of the op group. This

designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 4 for ISIZE(I), where I is in the range ($121 \leq I \leq 150$), represents the designation of company for red and blue adjacent units represented by those unit numbers.

The term company is also associated with the command and control panel button selections. When a value equal to 4 is designated in array ISIZE for a given unit, the unit is displayed when the company button is selected.

Those units designated a size of 4 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 4 used to represent company.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.37 COMPUTED EQUIPMENT MOVEMENT RATE

3.37.1 English-language Definition

a) Term Name:

Computed Equipment Movement Rate

b) Term Description:

All equipment within a unit are separated by type. Movement rates are computed each time step for every type of equipment contained in the unit (a unit may have at most fourteen different types of equipment). This computation takes into account the following:

1) the distribution of the equipment type over its modes of operation
2) the input-defined desired speeds of the equipment type while operating in its various modes of operation
3) the amount of suppression experienced by the unit
4) environmental constraints

Two intermediate rates are computed for each equipment type. The rate of movement ultimately assigned is the smaller of the two intermediate results.

The first intermediate rate is the average rate, involving the several speeds associated with the equipment operating in several different modes. The average is weighted appropriately by the fraction of equipment operating in each mode.

The other intermediate rate, is the maximum movement rate achievable by the equipment type in the terrain in which it is situated. This maximum rate is derived by applying environmental degradation factors to the top operational speed (independent of operating mode) of the equipment type. Note that this maximum rate may in fact be smaller than the average rate computed, particularly when environmental degradation effects are pronounced. The final movement rate assigned to the equipment type is determined by the smaller of the two intermediate rates.

c) Term References:

Computed equipment movement rate is determined by the ground fire module every time step. Each equipment type within the unit has a movement rate determined. The equipment type yielding the slowest computed movement rate determines the movement rate of the unit.

3.37.2 Programmatical Definition

a) Term Name:

AVGVEL (same as TCTR), VELMAX (same as TCTRMX)

b) Term Description:

The FORTRAN variable AVGVEL (and TCTR) is temporary storage for the first intermediate movement rate. The FORTRAN variable VELMAX (and TCTRMX) is temporary storage for the second intermediate movement rate. Two variable names are used for each intermediate rate because they are used by different subroutines.

AVGVEL,TCTR = average rate of movement for an equipment type, weighted by the fraction of the equipment operating in each of its modes

VELMAX,TCTRMX = absolute top speed attainable by an equipment type in the terrain in which it is situated

These variables are model generated variables that are used in the Ground Fire Module.

(Note that the computed movement equipment rate is given by AMIN (AVGEL,VELMAX)).

c) Term References:

The FORTRAN variables AVGVEL(TCTR) and VELMAX(TCTRMX) are temporary storage used for each equipment type within the unit. They record the two intermediate speeds involved in determining the final speed. The final rate is compared with the rates of other equipment types in the unit. If it is the slowest, the unit movement rate is set to this rate; otherwise the next equipment type is examined, and intermediate rates are computed and stored into the variables AVGVEL and VELMAX again. The variables AVGVEL,TCTR,VELMAX, and TCTRMX are

used in the following subroutines:

AMOVUL
FIRALO
NOTGT
ORGFIR
SPTALO
WTSUB

3.38 CONTROL MEASURE

3.38.1 English-language Definition

a) Term Name:

Control Measure

b) Term Description:

Control measures are points, lines, or areas within the battlefield which have special significance with respect to unit movement and firing. Control measures serve to reduce confusion and aid in coordinating operations of independent units within a given army. For instance, certain control measures are specified to prevent friendly units from (unknowingly) firing upon other friendly units; control measures also partition the area of operation into regions and assigns responsibility of these regions to various units. Thus if one unit crosses a control measure and encroaches into another unit's region, an alert must be generated to warn that the encroaching unit is a friendly unit.

Control measures are modeled as series of connecting straight lines, which may or may not close to form polygons. Area control measures are represented by polygons having not more than eight sides. Linear control measures comprise of no more than seven connecting line segments. Point control measures are represented by small squares (100 meter sides) centered about the X-Y coordinates specifying the point; thus point control measures are in reality area control areas.

A maximum of 100 control measures may exist in the model at any one time. Control measures may be defined by input or created (and deleted) interactively with the control measure menu. Each control measure is associated with a unit, and may affect one or both (i.e., red and blue) armies. Some control measures exist for display purposes only, whereas others will cause alert messages to be generated. For flexibility in displaying measures, the associated "unit" may be an operational grouping or a nonexistent (fake) unit. The unit

numbering convention is as follows:

1. Unit numbers 1-100 are legitimate units.
2. Unit numbers 101-120 correspond to operational groupings 1-20.
3. Unit numbers 121-135 are reserved for fake red units.
4. Unit numbers 136-150 are reserved for fake blue units.

Alert messages will be generated for control measure crossings as follows:

1. If the unit number associated with the measure is between 101 and 120 inclusive, then alerts will be generated only for units belonging to the corresponding operational grouping.
2. If the unit number associated with the measure is less than 101 or greater than 120, then alerts will be generated only for that unit (which is a fake unit if the unit number is greater than 120) and its subordinate units, as defined by next higher command data.

Presently the CATTS math model contains 37 different types of control measures; however the math model can accommodate up to 255 different types. The different types of control measures are classified into six subsets (called control measure classes):

1. lines
2. areas
3. zones
4. positions
5. bases
6. points

Control measures are distinguished by type and class for display and alerts-generation purposes.

c) Term References:

Control measures is referenced by specifying the control measure number, which is an integer ranging from 1 to 100 inclusively. Control measures are associated with units or operational units, and cause alert messages to be generated, depending upon the behavior of the units (or operational groupings) with respect to the control measures. Both movement and firing logics use

control measures to restrict or guide the maneuver and firing activities of units. Interactive command and control references control measures through the control measure menu. This menu allows control measures to be created, deleted, or moved elsewhere in the battle area. Note that control measures are automatically deleted whenever its associated units or operational groupings become defunct.

3.38.2 Programmatical Definition

a) Term Name:

ICM(15,000)

b) Term Description:

The integer array ICM is both a data base input variable and an interactively generated variable. As a data base input variable, it is defined in the control measure input deck, and, as an interactively generated variable, it is created from the control measure menus (3 different menus, one for points, one for lines, and one for areas). It contains packed data describing each control measure. The array allows for 100 control measures to be specified. Fifteen words of data is reserved for each control measure; these words contain the following information:

1) Control measure type
2) Unit number associated with the control measure
3) Army (red, blue, or both) affected by the control measure
4) A display flag
5) A unique name
6) Data specifying the X-Y coordinates of the points defining the line segments comprising the control measure

(J,IKM) Definition of IKMth control measure

J=1 Code word. Zero means measure not in use

Otherwise byte packed as follows:

Byte 0 = type number of this control measure

Byte 1 = unit number to which measure applies
Also applies to all subordinate units.

Byte 2 = 0 means red and blue measure

1 means red measure

2 means blue measure

Byte 3 = 0 means measure for display purposes only

N.LT.64 means generate control measure alert type N whenever a unit to which measure applies moves across measure.

N.GE.64 and .LT.128 means generate control measure alert type (N MOD 64) whenever a unit to which measure applies fires support fire weapons across measure.

N.GE.128 means generate control measure alert type (N MOD 64) whenever unit to which measure applies fires support fire weapons short of measure

Note that firing violations will only occur when a command and control fire command orders fire which violates a control measure.

J=2 Minimum X coordinate of any point on control measure

J=3 Maximum X coordinate of any point on control measure

J=4 Minimum Y coordinate of any point on control measure

J=5 Maximum Y coordinate of any point on control measure

J=6 TO J=13 Words defining X,Y coordinates of up to 8 to points defining control measure. A value of -1 means there are no more points for this measure. Each word is packed in half words as follows.

HW 0 + ICM(2,IKM) gives X coordinate of (J-5)th point on this measure

HW 1 + ICM(4,IKM) gives Y coordinate of (J-5)th point on this measure

Note the following assumptions:

Areas always have last point connected to first point.

Control points are modeled as small, square control areas centered at control point. Coordinates of square are given as ICM(6,IKM) ICM(7,IKM),ICM(8,IKM),ICM(9,IKM). Then ICM(10,IKM) = -1 to indicate no more points. True coordinates of point are given by ICM(11,IKM) ICM(12,IKM)

J=14 EBCDIC representation of the 8 character
15 name assigned to control measure IKM

c) Term References:

The integer array ICM, which contains data describing each control measure is indexed by word number J, $1 \leq J \leq 15$ and control measure number IKM, $1 \leq IKM \leq 100$. Each control measure IKM references fifteen words of storage containing information describing the control measure. The array ICM is used by the following subroutines:

ACTIVATE	CONTMS
CK4XING	FORRAM
CMINP	LEAVOG
CMSEGMNT	NEWFWDUN
CNTRLMSR	RMVOPGP

3.39 CONTROL MEASURE CLASS

3.39.1 English-language Definition

a) Term name:

Control Measure Class

b) Term Description:

A control measure class is a disjoint subset of the set of control measure types. Currently the CATTS math model contains 37 different types of control measures (see term definition control measure type). Each of these types can be categorized into one of seven classes:

1) line
2) area
3) zone
4) position
5) base
6) point
7) All others not included in 1 through 6.

Control message type classification is required to distinguish messages which must be composed for different types of control measure alerts.

c) Term References:

Control measure class is referenced when ever a control measure alert must be constructed and sent out.

3.39.2 Programmatical Definition

a) Term Name:

ICMBREAK(6), LABELCM(6)

b) Term Description:

The integer arrays ICMBREAK and LABELCM are model generated variables that are used in subroutine FORMAIN (data statement). They contain data describing the attributes of control measure class. ICMBREAK contains pointers into the array NAMECM. The pointer references elements (i.e., indices) which separate each control measure class. The array LABELCM contains alphanumeric character data spelling the names of each control measure class. Note that six names have been defined. The last class of control measures does not require a name. These names are used only to generate appropriate alert messages.

ICMBREAK(ICMCL)	The index (using the background indexing scheme) pointing to the control measure type which marks the breakpoint between control measure class ICMCL and class ICMCL+1.
LABELCM(ICMCL)	The four character names for the CIMCLth class of control measure - LINE,AREA,ZONE,POSN,BASE, PNT.

c) Term References:

The arrays ICMBREAK and LABELCM are indexed by class number ICMCL where $1 \leq \text{ICMCL} \leq 6$. These arrays are used by the following subroutines:

CK4XING
FORMAIN

3.40 CONTROL MEASURE TYPE

3.40.1 English-language Definition

a) Term Name:

Control Measure Type

b) Term Description:

Presently the CATTS math model considers 37 different types of control measures. Each of the different types of control measure has a unique name type. They are listed below:

1. limit of advance
2. objective
3. FEBA objective
4. flight route corridor
5. supply dump
6. airfields
7. preplanned target
8. assault line
9. fire coordination line
10. line of contact
11. line of departure
12. phase line
13. air control line
14. boundary line
15. no fire line
16. probable line of deployment
17. delay line
18. assembly area
19. no fire area
20. area of support
21. restricted area
22. aircraft patrol area
23. prohibited flying area
24. vulnerable area
25. drop zone
26. landing zone
27. pickup zone
28. inner artillery zone
29. attack position
30. defense position
31. fire support base
32. patrol base
33. check point
34. coordination point
35. release point
36. start point
37. passage point
38. not used
39. not used
40. not used

Control measures are distinguished by type because different types serve different purposes. Some types exist just for display purposes whereas others will cause an alert message to be generated if the control measure type is violated.

c) Term References:

Control measure type is referenced by the type integer which presently is a number between 1 and 37 inclusively. In reality there exist two indexing scheme to reference control measure type: one scheme is used only by the foreground software and the other scheme is used only by the background software. The foreground scheme considers 121 different types of control measures, although to date, only 45 have actually been defined. The remaining 76 types have not been defined and their indices are reserved for future expansion. The background software utilizes 37 of the 45 types of control measures established by the foreground software. This presents conflicting indexing schemes. To resolve the conflict, a background subroutine maps the type numbers (given by the foreground) into consecutive integers ranging from 1 to 37 inclusively for use by the background software only.

Types of control measures must be distinguished because some are used only for display purposes while others require the name of the type to be composed in an alert message.

3.40.2 Programmatical Definition

a) Term Name:

NAMECM(3,40)

b) Term Description

The integer array NAMECM is both a data base input variable and an interactively generated variable. As a data base input variable, it is defined in the control measure input deck, and, as an interactively generated variable, it is created from the control measure menus (3 different menus, one ofr points, one for lines, and one for areas). It contains the character data spelling out the names of each type of control measure currently being modeled. The names consist of at most twelve alphanumeric characters. They are used to generate alert messages.

NAMECM(J,ICMT) = the 12 character name (J=1,3) for the ICMTth control measure type where $1 \leq ICMT \leq 40$.

c) Term References:

The array NAMECM is indexed by word numbers J, $1 \leq J \leq 3$, and type number ICMT, $1 \leq ICMT \leq 40$. The index ICMT is chosen by referencing a byte array MTYPOFCM. The control measure type number given by the foreground is used to reference a byte in the array MTYPOFCM. This byte contains an integer for the subscript ICMT (presently between 1 and 37 inclusively) which is used as an index for array NAMECM. The array NAMECM appears in the following subroutines:

CK4XING
FORMAIN

3.41 CORPS

3.41.1 <u>English-language Definition</u>

a) Term Name:

Corps

b) Term Description:

A corps is designated in the model by setting the size value for a specific unit to 9. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for corps is "XXX".

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a size of 9 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 9 used to represent corps.

3.41.2 <u>Programmatical Definition</u>

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit input deck for units 1-100, the operational group input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150.

A corps is designated for Unit I in the model by setting array ISIZE(I) to a value of 9 as part of this data base inputs. This value is never altered for units. A corps is designated for operational group (I-100) by setting array ISIZE(I), $(101 \leq I \leq 120)$, equal to 9 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a

result of interactively defining new op groups through the task organization menu. A value of 9 for ISIZE(I), where I is in the range ($121 \le I \le 150$), represents the designation of corps for red and blue adjacent units represented by those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \le I \le 100$) for normal units, ($101 \le I \le 120$) for op groups, and ($121 \le I \le 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.42 CURRENT LOAD

3.42.1 English-language Definition

a) Term Name:

Current load

b) Term Description:

The current load for personnel, equipment, gas and diesel fuel for ground units is input at initialization time from the data base and recalculated each minute to account for utilization, casualties, attrition, and the like. For air units, the current load is initially set at the time an air mission is created and recalculated each minute to account for utilization, casualties, attrition, etc.

c) Term References:

The arrays containing current load data are unit attributes and, as such, are referenced by unit number.

3.42.2 Programmatical Definition

a) Term Name:

PERS(I),TOTEQU(I,J),CLAVGAS(I),CLDIES(I),CLGAS(I)

b) Term Description:

PERS (I)	Number of personnel currently in Unit I.
TOTEQU (I,J)	Total number of pieces remaining of Jth equipment type carried by Unit I.
CLAVGAS (I)	The current load of aviation fuel for Unit I.
CLDIES (I)	Current load (aviation amount) of diesel fuel for Unit I(1-99) in gallons.
CLGAS (I)	Current load (present amount) of gas for each of Unit I (1-99) in gallons.

PERS and TOTEQU are both data base input variables (PERS is created from data input in MENNOW in subroutine UNINP) and model generated variables. As data base input variables, they are defined in the unit input deck, and, as model generated variables, they are used by the Ground Fire, Air, and Ground Movement Modules. CLAVGAS, CLDIES, and CLGAS are model generated variables. CLAVGAS is used by the Air Module, and CLDIES and CLGAS are used by subroutine USEFUEL.

c) Term References:

All current load variables are unit attributes and, therefore, their unit index, IU, ranges from ($1 \leq IU \leq 100$). The second index for equipment basic load (TOTEQU) ranges from ($1 \leq J \leq 14$) representing up to 14 different equipments that a unit can have. Subroutines using the basic load variables include the following:

ADJROM	INPUT
ADW	LOWALRT
AIREVENT	MOVE
AIRGRND	MOVMNT
AIRHOTZN	OBSDELAY
AIRMOV	ORGFIR
AIRMOV2	PERSLINE
AMOVUL	RADAR
BOTGTS	REDCON
CHGCRT	REDIST
CRDLIC	RESUPPLY
EFFNS	SAVE
ENGRSPT	SAVEINP
EQUPLINE	SPTALO
FIRAGEN	STATREP
FIREVNT	STEP
FIRSORT	SUPRES
FORRAM	UNINP
FUELLINE	USEFUEL
GENFIR	VISUAL
INIT	WPNEFF

3.43 DATA BASE

3.43.1 English-language Definition

a) Term Name:

Data Base

b) Term Description:

The CATTS data base consists of all the data (both letters and numbers) needed to initialize the CATTS mathematical model. This data is used during the execution of the mathematical model.

c) Term References:

(Not applicable)

3.43.2 Programmatical Definition

a) Term Name:

Data Base

b) Term Description:

The CATTS data base is contained in various files located on various disk areas as shown in Table 3-1. The files in Table 3-1 contain almost all of the data necessary to initialize the mathematical model. The remaining small amount of data necessary to initialize the mathematical model is in data statements, which are FORTRAN statements in the code itself. Data in these statements can be changed, however, it is more difficult to change than the data on the data base files. Changing numbers in the data statements involves changing the subroutine which contains the data statement, i.e., changing the data statement itself, then recompiling the subroutine which was just changed, and then reloading the model to incorporate this new subroutine into the load module. The amount of data contained in the data statements in the mathematical model is a small percentage

Table 3-1. CATTS Data Base Disk Areas

FILE NAME	DISK AREA	DESCRIPTION OF DATA ON FILE
GOLD SILVER ATTACK	DB DB DB	Equipment, weapon effects, unit type, unit, op state control, suppression, control measure, mine, obstacle, forticiation, road, and preplanned mission data for FEBA GOLD, SILVER, and ATTACK exercises, respectively. These are the baseline files.
NGOLD NSILVER	DB DB	Data is in same format as data on GOLD, SILVER, and ATTACK, but new data is being incorporated and evaluated to eventually become part of the next baseline.
BFBAGOLD BFBASLVR BCORDATT	DB DB DB	Binary equivalent of the data on NGOLD, NSILVER, or NATTACK, respectively, blocked in records 16,384 words long. These files are filled by the computer.
EFBAGOLD EFBASLVR ECORDATT	DB DB DB	Prescheduled events files for the respective exercises. The command and control on these files will occur everytime the particular exercise is run.
PFBAGOLD PFBASLVR	DB DB	Preplanned mission files for the respective exercises. The computer fills these files from information on the preplanned mission portion of files NGOLD, NSILVER, and NATTACK, respectively, when the procedure to make the binary files (BFBAGOLD, etc.) is executed.
RELIEF	DA	Elevation data for the game area. Elevation for 25M by 25M grids is packed, i.e., more than one piece of data is stored in a 32 bit word. The information on this file is used in array IELEVE (34,472).
ELEVDISP	DA	Elevation data used in conjunction with RELIEF. This data file is needed to allow the packing of RELIEF and fills array IEVDSP (3,750).
VEGCOMP	DA	Contains information about the vegetation types, sizes, classes, etc., in the game area. Used in arrays H(16,4), HB(9), RHO(16,4), RMAX(16,4), W(16,4).

Table 3-1. CATTS Data Base Disk Areas (Continued)

FILE NAME	DISK AREA	DESCRIPTION OF DATA ON FILE
VEGLOC	DA	Vegetation location in the game area. The area occupied by a vegetation class is described by a collection of rectangles, circles, and triangles. The data from this fills arrays ICL(225), ITRC(225), XPOLY(5,225), YPOLY(5,225), IVEGLOC(2,64).
SOIL	DA	Soil type location in the game area. Information is 75M apart in Y direction and 25M apart in X direction and is packed (i.e., more than one number per 32 bit word). Used in arrays ISOILE(361), JSOILE(15687).

of the total data in the data base. Most of this data need not be moved to the data base for any reason. However, to make it easier to change this data, for example, the air to ground weapon effects, some of the data in the data statements eventually may be moved onto the disk file base, i.e., NGOLD, NSILVER, and NATTACK.

c) Term References:
(Not applicable)

3.44 DEPLOYMENT (OPERATIONAL GROUPING)

3.44.1 English-language Definition

a) Term Name:

Deployment (Operational Grouping)

b) Term Description:

Deployment refers to the movement of units within an operational grouping to designated locations relative to the forward-most (i.e., controlling) unit of the operational grouping. An operational grouping is deployed if all its units have arrived at their positions; otherwise the operational grouping is said to be in the process of deploying.

Every unit receives its deployment location relative to the lead unit. This location is given by a pair of numbers; the first number indicates the forward distance of the unit from the lead unit (positive number indicates distance in front of lead unit, negative number indicates distance behind); the second number specifies the lateral distance (positive number indicates distance to the left of lead unit, negative number indicates distance to the right). By convention, the direction of movement of the lead unit is the unit's positive forward direction.

An operational grouping's deployment status can be represented by one of four possible states. This includes: not deployed and unable to move, not deployed and moving in column formation, technically deployed, and actually deployed.

c) Term References:

The deployment status is an attribute of operational grouping; it is referenced by the operational grouping number, which is an integer from 1 to 20 inclusively. Deployment status is used by the engagement logic and the movement logic to position units for encounters with the enemy. This involves establishing engagements, FEBA's frontages, etc.

3.44.2 Programmatical Definition

a) Term Name:

IDPCOD(20)

b) Term Description:

The integer array IDPCOD is a data base input variable, an interactively generated variable, as well as a model generated variable. As a data base input variable, it is defined in the operational group input deck; as an interaciively generated variable, it is created from the maneuver menu; and, as a model generated variable, it is used in subroutines ARRIVE, CHGOPN, LEAVOG, NEWENG, NEWFWDUN, NEWMOV, OGDIR, OGLOC, RMVOPGP, and TASKORG. It contains a deployment code for each operational grouping. The code can range from zero to three inclusively, and changes as situations arise during the simulation.

IDPCOD(IOPG) = deployment code of the IOPGth ($1 \leq$ IOPG ≤ 20) operational grouping:

0 - units of operational grouping are not deployed and cannot move

1 - units of operational grouping are not deployed, moving in column formation

2 - units of operational grouping are "technically" deployed which means that the units have all arrived at their designated positions relative to the forward-most unit, but the units are

moving together in this deployed formation towards a specific destination to engage the enemy; the movement code of the operational grouping must be 6,8,9,10,11,12,13, or 14.

3 - units of operational grouping are "actually" deployed which means that the units have all arrived at their designated positions relative to the forward-most unit; the movement code of the operational grouping must be 1,2,3,4,15, and 16.

c) Term References:

The array IDPCOD is indexed by operational grouping number IOPG, where IOPG is an integer ranging from 1 to 20 inclusively. The array is used by the following subroutines:

ADJDIR	OGDIR
ARRIVE	OGINP
CHGOPN	OGLOC
DIRMOV	OGLOC2
LEAVEOG	OG2UNI
MANEUVER	PREMOV
NEWENG	RMVOPGP
NEWFWDUN	STATREP
NEWMOV	TASKORG

3.45 DETECTION

3.45.1 English-language Definition

a) Term Name:

Detection

b) Term Description:

In the CATTS mathematical model, the occurrence of detections between: a) ground units and b) ground and air units is modeled. Detections between ground units occur if it is determined that one unit is eligible to detect another, and the probability of visual detection, radar detection, aural detection, or detection by unattended ground sensor field exceeds input thresholds. Detections between ground and air units occur if it is probabilistically determined that a ground unit can visually detect an enemy air unit or that an air unit, with its sensors, can detect an enemy ground unit. Many environmental and tactical considerations and a wide range of sensor types have been included in the detection models.

c) Term References:

(Not applicable)

3.45.2 Programmatical Definition

(Not applicable)

3.46 DIRECTION OF MOVEMENT

3.46.1 English-language Definition

a) Term Name:

Direction of Movement

b) Term Description:

The direction of movement of a unit or operational grouping is given by the sine and cosine of the angle measured counter clockwise from the positive X-axis with respect to the fixed frame of reference. The fixed frame of reference is the right-handed, rectangular Cartesian coordinate system (i.e., due east is zero degrees, due north is ninety degrees, etc.). This angle measuring convention is adopted by the CATTS background software.

The initial direction of movement for a unit or operational grouping is input as an angle measured in degrees with respect to the above mentioned coordinate system. This angle is translated to radian measure and immediately converted and stored into memory as the sine and cosine of the angle. All subsequent references to direction of movement are in terms of sine and cosine of the angle.

Direction of movement chosen interactively on menus from the controller console utilize a different measuring convention. Angles are measured clockwise from the positive Y-axis (i.e., due north is zero degrees, due east is ninety degrees, etc.) This is consistent with the standard compass measuring convention; angles chosen in this manner must be transformed to follow the CATTS angle measuring convention. Thus, when an angle is chosen interactively, it must be translated to an angle measured counterclockwise from the positive X-axis, then converted and stored into memory as the sine and cosine of the angle.

c) Term References:

Direction of movement is an attribute of unit and operational grouping. It is referenced by many functions of the math model, but is used mainly by the movement, firing, and engagement functions.

3.46.2 Programmatical Definition

a) Term Name:

PDIR(100,2) unit direction of movement
DPMVT(20,2) operational grouping direction of movement

b) Term Description:

The direction of movement for a unit is stored in the following FORTRAN variables:

PDIR(IU,J) = Sine (J=1) and cosine (J=2) of angle of direction faced by unit *IU*, using the Cartesian coordinate system of angle measurement.

The direction of movement for an operational grouping is stored in the following FORTRAN variable:

DPMVT(IOPG,J) = Sine (J=1) and cosine (J=2) of angle of direction faced by operational grouping IOPG, using the Cartesian coordinate system of angle measurement.

In both cases, the initial angles are input in degrees and converted into sine and cosine. The direction of movement is re-computed each time step, for all units and operational groupings.

PDIR and DPMVT are data base input variables, interactively generated variables, as well as model generated variables. PDIR and DPMVT, as data base input variables, are defined in the unit input deck and operational grouping input deck, respectively. As interactively generated variables, they are created from the maneuver, tasking organization, and unit location menus. As model generated variables,

they are used in the Ground Movement and Engagement Modules, with PDIR also used in the Command and Control Module.

c) Term References:

The variable PDIR is indexed on unit number, represented by IU in the nomenclature list ($1 \leq IU \leq 100$). PDIR is used in the following subroutines:

ADJDIR	MANEUVER
ADW	MOVE
AIRABORT	MOVMNT
AIREVENT	NEWENG
AIRGRND	NEWMOV
AIRHOTZN	OGDIR
AIRMOV	OG2UNI
AREA	OPPLAN
ARRIVE	OVERUN
CRDLOC	POINTGT
DIRMOV	RADAR
ENCTR	TASKORG
FINDWA	UNBLOK
FORMAIN	UNINP
FORSIGI	WTHDRW

The variable DPMVT is indexed on operational grouping number, represented by IOPG in the nomenclature list ($1 \leq IOPG \leq 20$). DPMVT is used in the following subroutines:

ADJDIR	OGDIR	STATREP1
ARRIVE	OGINP	TASKORG
CHGOPN	OGLOC	UN2FEB
DEPLOC	OGLOC2	
DIRMOV	OG2UNI	
MANEUVER	PREMOV	
NEWENG	REL2FWDU	
NEWFWDUN	SETOGCDS	
NEWMOV	STATREP	

3.47 DISPERSION FACTOR

3.47.1 English-language Definition

a) Term Name:

Dispersion Factor

b) Term Description:

Dispersion factor is a calculation performed for artillery weapons using weapons effects function 6. The result of the dispersion factor calculation represents an estimate of the number of rounds actually falling within the target unit area. The following calculation is performed:

$$F_{\alpha} = 1 - \exp\left[-A/2\pi\sigma^2(R)\right]$$

where F_{α} = fraction of rounds falling within target unit area
$\sigma(R) = C_1 + C_2R + C_3R^2$, where C_1, C_2, and C_3 are input coefficients and R is range to target unit
A = area of target unit

Only those rounds falling within the target area are evaluated for casualty and damage effect.

c) Term References:

A list of weapons requiring a dispersion factor calculation is specified from data base inputs and stored. A set of three coefficients is also specified from data base inputs corresponding to each weapon. Weapon (equipment) number, therefore, determines which set of coefficients is to be used in the dispersion factor calculation.

3.47.2 Programmatical Definition

a) Term Name:

The term dispersion factor is represented programmatically by two arrays, IDISPR(I) and DISPER(I).

b) Term Description:

IDISPR (I)	Weapon type for which effect function 6(firing) is used ($1 \leq = I \leq = 10$)
DISPER (I,J)	Constants for use in calculating fraction of rounds falling on the target unit where I is the index from IDISPR ($1 \leq I \leq 10$), J is the number of constants used ($1 \leq J \leq 3$)

IDISPR and DISPER are data base input variables that are defined in the weapon effects input deck.

c) Term References:

The equipment number is matched against the entries in IDISPR (I) to determine the appropriate index I, into the DISPER(I,J) table. All three entries are used (J = 1,2,3) in the dispersion formula in subroutine RDSON. Input values for the two arrays are read in subroutine WEFINP.

3.48 DIVISION

3.48.1 English-language Definition

a) Term Name:

Division

b) Term Description:

A division is designated in the model by setting the size value for a specific unit to 8. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for division is "XX".

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a size of 8 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 8 used to represent *division*.

3.48.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit input deck for units 1 - 100, the operational group input deck for units 101 - 120, and the higher and adjacent unit input deck for units 121 - 150.

A division is designated for Unit I in the model by setting array ISIZE(I) to a value of 8 as part of the data base inputs. This value is never altered for units. A division is designated for operational group (I-100) by setting array ISIZE(I),($101 \leq I \leq 120$), equal to 8 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 8 for ISIZE(I), where I is in the range ($121 \leq I \leq 150$), represents the designation of division for red and blue adjacent units represented by those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.49 ENGAGEMENT

3.49.1 English-language Definition

a) Term Name:

Engagement

b) Term Description:

An engagement in CATTS is a confrontation of a set of Red and Blue units in which the units involved direct their attention primarily toward one another. However, two opposing units are not prohibited from firing at each other when either or both are unengaged. Likewise, two opposing units in different engagements are not prohibited from firing at each other.

A new engagement is formed if the following criteria are satisfied:

- no obstacle blocks the engaging unit
- the engaging unit must be presently unengaged
- the engaging unit must not be airborne
- the engaging unit must not be under maneuver control, or, if he is, must have been commanded to seek engagement
- the engaging unit must have visually detected the opposing unit he's attempting to engage with
- the units must be within a pre-defined "must fight" distance associated with the engaging unit's type, or the units are within the engagement range pre-defined for the unit type (but not within the "must fight" range) and neither unit strongly objects to engaging (as indicated by his travel code).

If these criteria are satisfied, the following parameters are calculated, which combine to define the engagement:

- the direction of the engagement axis
- the half-frontage of the Blue and Red forces
- the X,Y coordinates of the center point of the Blue and Red FEBA's
- position of the Blue and Red FEBA's for their respective directions of movement
- the area of close support for both Blue and Red, to be used to direct opposing artillery fire
- the number of the controlling Blue or Red operational grouping, if any.

The formation of engagements affects the ground movement and ground firing logic in the model. If units are moving toward a destination, their direction and destination are altered if the conditions for engagement are satisfied. The unit may take a specially-deployed position within the engagement. The ground fire logic uses engagements in distinct ways for direct fire and support fire. An opposing unit in the same engagement may be determined ineligible for direct fire if he has been removed from combat status in the engagement.

For automatic support fire allocation, an engagement has close support and interdictory regions defined within it. Any opposing unit within the close support region is fired at from mode 3 or 4, the close support modes. Any opposing unit within the interdictory region is fired at from mode 5.

c) Term References:

Engagements are model entities whose attributes consist of numerical values assigned to the program variables listed below.

3.49.2 Programmatical Definition

a) Term Name:

DIREAX (JENG, J)
IBFRNT (JENG)
IFEBAB (JENG, J)
IRFRNT (JENG)
IXPHIB (JENG)
IXPHIR (JENG)
KSAREA (JENG, JCOLOR)
NCBOG (JENG)
NCROG (JENG)

b) Term Description:

The following program variables are model generated variables that are used by the Engagement Module. They are set when an engagement is formed, and together combine to define the engagement:

DIREAX (JENG,J)
Direction of the JENG-th engagement axis, Blue to Red. Stored as sin (J=1), cos (J=2).

IBFRNT (JENG)
Half-frontage of Blue force in the JENG-th engagement.

IFEBAB (JENG,J)
The X (J=1) and Y (J=2) coordinates of center point of Blue FEBA in the JENG-th engagement.

IFEBAR (JENG,J)
The X (J=1) and Y (J=2) coordinates of center point of Red FEBA (location of controlling Red OG or intersection of Red FEBA with JENG-th engagement axis).

IRFRNT (JENG)
Half-frontage of Red force in the JENG-th engagement.

IXPHIB (JENG)
Position of Blue FEBA in the direction of Blue movement (= -9999999 if engagement no longer exists) of the JENG-th engagement.

IXPHIR (JENG)
Position of Red FEBA in the direction of Red movement (= 9999999 if engagement no longer exists) of the JENG-th engagement.

KSAREA (JENG,JCOLOR)
Area of close-support target region in the JENG-th engagement; Red firing against Blue region (JCOLOR=1), Blue firing against Red region (JCOLOR=2).

NCBOG (JENG)
For the JENG-th engagement, number of controlling Blue operational grouping (zero if none).

NCROG (JENG)
For the JENG-th engagement, number of controlling Red operational grouping (zero if none).

Engagements are formed as ground unit positions are processed. No engagements are defined either by data base input or interactive controller commands. All engagements are formed via the ground movement logic updating unit positions.

c) Term References:

All engagement arrays are indexed by engagement number, JENG ($1 \leq JENG \leq 12$). If a thirteenth engagement is attempted, the exercise will be stopped.

Subroutine NEWENG is the principal routine used to create new engagements.

Subroutines FIRELG and CLSPTG are the principal ground fire routines using the engagement information, and subroutine ENCTR is the principal routine to initiate changing movement codes, directions, and destinations for units coming within engagement range of the enemy.

3.50 ENGAGEMENT FRONTAGE

3.50.1 English-language Definition

a) Term Name:

Engagement Frontage

b) Term Description:

In the CATTS simulation of an engagement, engagement frontage is defined to be the distance perpendicular to the engagement axis spanning the width of the battle area. The frontage of a side of the engagement is either the unit width, if a unit is entering the engagement for that side, or the lateral distance of the operational group if it was previously unengaged. The lateral distance is determined by the unit in the operational group which has the largest perpendicular distance from the engagement axis. The engagement frontage of the side determined to be the non-initiator of the engagement is expanded or contracted based on table entry inputs at model start time. Note that engagement frontage, in conjunction with the X-Y coordinates of a FEBA location (see term definition of FEBA) form the imaginary line constituting the FEBA. Generally an engagement consists of two frontages, one for the Red forces and one for the Blue forces. However, there are instances when an engagement will contain only one. If an already-engaged operational group or unit becomes involved, as a non-initiator, in another engagement (i.e., it is being attacked by another enemy force thus forming a one-sided engagement), it's engagement frontage is set to zero indicating a nonexistent frontage.

c) Term References:

Engagement frontage is an attribute of the entity engagement, hence it can be referenced by engagement number. When two opposing forces are in confrontation, their engagement frontages are used to determine their respective forward edges of the battle area (FEBA's). The establishment of FEBA's provide parallel reference lines used by the movement and firing logic to perform engagement decisions and operations which simulate the battle.

Specifically, units are deployed, or moved to certain deployment positions where they can utilize direct fire weapons against one another.

3.50.2 Programmatical Definition

a) Term Name:

IBFRNT(12), IRFRNT(12)

b) Term Description:

Engagement frontages are distances which can be computed from values stored in the FORTRAN integer arrays IBFRNT and IRFRNT. The values stored in each of the above arrays are in half distances (i.e., half-frontages) describing each engagement frontage. Every engagement usually has two frontages associated with it (a Red frontage given in the array IRFRNT, and a Blue frontage given in the array IBFRNT). One-sided engagements are possible when an operational grouping or unit forms a new engagement to attack an already-engaged enemy. In these instances, the attacking force will have a frontage, but the engaged enemy will not (i.e., zero distance frontage).

IBFRNT (JENG)	half-frontage of Blue force in the JENG-th engagement.
IRFRNT (JENG)	half-frontage of Red force in the JENG-th engagement.

These variables are model generated variables that are used by the Engagement Module.

c) Term References:

The integer arrays IBFRNT and IRFRNT are indexed by the engagement number JENG, where JENG is an integer ranging in value between 1 and 12 inclusively. Note that a maximum of twelve engagements may exist concurrently. Generally the above arrays are initialized to zero each time step; frontages of zero distance indicate non-existent frontages. But as engagements are formed and maintained, subroutines NEWENG and LATDST compute and update the half-frontage values for the respective forces and stores these values in the

appropriate arrays above. For a given engagement, the half-frontage value is either half the unit width if only a unit is initiating the engagement, or in the case where an operational group is the initiator, the lateral distance determined by the unit of the group having the largest perpendicular distance from the engagement axis plus half of that unit's width. The arrays IBFRNT and IRFRNT are referenced by the following subroutines:

ADJIR
CLSPTG
DEPLØY
FINDFØ
FØRRAM
FRNTGS
DATDST
NEWENG
NGARGN

3.51 ENGINEERING SUPPORT

3.51.1 English-language Definition

a) Term Name:

Engineering Support

b) Term Description:

Engineering support is modeled as a set of reduction factors to be applied to the delay time suffered by units which have encountered obstacles. The amount of reduction in delay time depends upon the type of obstacle halting the unit. In order to qualify to use the reduction factors, at least one engineering unit must be present within the army.

A unit which has had its delay time reduced is considered to be engaged in an engineering task. The engineering task exists as long as the reduced delay time is in effect. When the reduced delay time has elapsed, the unit can start moving across the obstacle. At this point, the engineering task is considered complete and no longer exists. Keeping track of the number of tasks is important because the model only allows ten engineering tasks to occur concurrently for an army. The completion of one task allows another unit to receive engineering support.

Engineering tasks are allocated on the basis of longest delay time. Those units suffering the longest delay will receive engineering support first. The model attempts to grant all request for engineering support when possible. Once the maximum number of concurrent tasks have been achieved, additional engineering support requests are denied, until the number of tasks is reduced below the maximum.

c) Term References:

Obstacle type is used to reference the reduction factors representing engineering support. Reduction factors are used by the movement function to simulate the effects of delay reduction from engineering support.

3.51.2 Programmatical Definition

a) Term Name:

ENGRFCTR(10)

b) Term Description:

ENGRFCTR(IOBST) is a data base input variable that is defined in the namelist input deck. It is a fraction, between zero and one inclusively, used to reduce the delay time assessed against a unit stopped by an obstacle of type IOBST ($1 \leq IOBST \leq 10$); this factor is applied only if engineering support is available to the delayed unit. These factors must be pre-defined by NAMELIST input.

c) Term References:

ENGRFCTR is indexed on obstacle type, represented by IOBST ($1 \leq IOBST \leq 10$) in the nomenclature list. This variable is used in the following subroutine:

ENGRSPT

3.52 ENVIRONMENTAL DEGRADATION FACTORS

3.52.1 English-language Definition

a) Term Name:

Environmental Degradation Factors

b) Term Description:

Environmental degradation factors are a set of fractions ranging in value from zero to one inclusively. They are computed every minute during the simulation to model the retarding effect of terrain on the movement rate of a given self-propelled equipment type. When a unit is traveling off-road, a fraction is derived for each of the following environmental features:

1. relief (slope)
2. vegetation
3. soil type
4. micro-relief
5. visibility

Onroad travel considers only the following:

1. relief (slope)
2. visibility
3. road type
4. damage assessed the road by air and ground ordnance.

If the unit is moving in a dismounted mode, a fractional factor is also computed to model the effects of human performance degradation. The set of fractions derived are used as multiplicative factors to determine one overall fraction which will represent the combined effects of the various environmental features. An overall factor is computed for each self-propelled equipment type within the unit. This combined factor is applied to the maximum operational speed of an equipment type within the unit. This produces a degraded top speed attainable by that equipment type.

c) Term References:

Degradation factors are computed for every self-propelled equipment type within a unit. The degradation factors are used in the computation of unit movement rate, which is done by the firing function. Each factor is referenced by the unit number and the position of the equipment type in the unit's equipment list.

3.52.2 Programmatical Definition

a) Term Name:

IDEGRAD(350)

b) Term Description:

IDEGRAD is a model generated variable used in subroutine ADJROM. It is a byte packed array containing the overall terrain degradation factor for each self propelled equipment type in all units. This factor is expressed as an integer (between zero and one hundred) obtained by converting the overall degradation factor into a percent. The computation is dependent on values stored in the following FORTRAN variables:

Offroad movement

1. SLOPET(I) - The instantaneous terrain slope for Unit I ($1 \leq I \leq 100$)
2. IVGSL(I) - The vegetation (first half word) and soil (second half word) classes for Unit I.
3. DFMIRF - Global movement degradation factor caused by overall micro-relief.
4. VISM - Global meteorological visibility (meters).

Onroad movement

1. SLOPET(I) - The instantaneous terrain slope for Unit I ($1 \leq I \leq 100$).
2. VISM - Global meteorological visibility (meters).
3. IRDTYPE(IRD) - Type of road IRD ($1 \leq IRD \leq 50$).
4. RDDMGE(RDSG) - Fraction of road segment IRDSG damaged by air or ground fire ($1 \leq IRDSG \leq 500$)

c) Term References:

IDEGRAD is referenced by the unit, and the placement in the unit's equipment list. To obtain the degradation factor for the Jth equipment in Unit I, reference the Kth byte from IDEGRAD(1), where K = (I-1) + 14 + J-1. The index J ranges from 1 to 14 inclusively and the index I ranges from 1 to 100 inclusively. IDEGRAD is used by one or more of the following subroutines:

ADJROM
AMOVUL
CRGFIR

3.53 EQUIPMENT CLASS

3.53.1 English-language Definition

a) Term Name:

Equipment Class

b) Term Description:

Equipment class has two distinct uses. For ground equipment, it is used as a descriptor of the equipment's vulnerability to air ordnance, which is a parameter in assessing damage from delivery of the ordnance. For air ordnance, equipment class provides a convenient index into the air-delivered-weapons data arrays.

c) Term References:

Equipment class is an equipment attribute, indexed directly by unit number. It is used in the air-developed-weapons function to determine the correct delivery and lethality data to be used for an air ordnance against a ground equipment.

3.53.2 Programmatical Definition

a) Term Name:

IEQCLS(IEQ)

b) Term Description:

IEQCLS (IEQ) For each ground equipment type IEQ, a general classification for air ordnance, defined as follows:

1 = SMALL ARMS
2 = LIGHT MORTARS
3 = ARTILLERY
4 = NON-ARMORED VEHICLES
5 = ARMORED VEHICLES

For air ordnance, an index into the adwdata arrays; not used for aircraft.

IEQCLS(IEQ) is a data base input variable that is defined in the namelist input deck.

c) Term References:

IEQCLS(IEQ) is an equipment attribute; hence, the range of IEQ is $(1 \leq IEQ \leq 80)$. Ground equipments range from $(1 \leq IEQ \leq 21)$ and $(26 \leq IEQ \leq 50)$. Air equipments range from $(22 \leq IEQ \leq 25)$ and $(69 \leq IEQ \leq 80)$. Since the array has no meaning for aircraft and air sensors, IEQCLS(51) through IEQCLS(68), inclusive, are not used. The following subroutines use this data:

ADW

ADWDATA

The data is read into this array as a namelisted input.

3.54 EQUIPMENT TYPE

3.54.1 English-language Definition

a) Term Name:

Equipment Type

b) Term Description:

Equipment type in the CATTS math model refers to material, machinery, weapons, and instruments required for a unit to operate. The availability of various types of equipment within a nunit dictates the functions (i.e., movement, detection, firing, etc.) that can be performed by the unit (provided enough personnel exist to man the equipment). Equipment type is modeled by determining the most important attributes common to all types; this provides the data structure (i.e., arrays) to be initialized in order to define a specific equipment. Presently, a maximum of eighty different types of equipment may be defined. Each unit is allowed to have up to fourteen different equipment types in its arsenal. Equipment type assignment is done by the user when units are being defined. Note that eighty different equipment types are available for the user to assign to units, and nothing prevents the user from assigning the same equipment type (though this may be unrealistic) to both red and blue units.

Every equipment type can be thought of in terms of one of four categories: nonweapons, direct-fire weapons, indirect-fire weapons, or support fire weapons. Examples of nonweapons include trucks, cargo aircraft, TVS-4 NOD (infrared night vision device), etc. Direct-fire weapons normally include rifles, machine guns, anti-tank guns, and other flat-trajectory armaments. The distinction between indirect-fire and support-fire weapons is not a sharp one; it is generally dependent on how the weapon is used. Indirect-fire weapons belong to, or move along with maneuver units to directly support the operations of these units (e.g., mortar, anti-tank guided missles). Support-fire weapons (e.g., 105-mm howitzers) usually remain stationary and support any and all units at various times during a battle.

Every equipment type can operate in several different modes. A mode of operation is the specific manner in which an equipment utilized with respect to the following:

1. rate of movement
2. ammunition type and rate of fire (if a weapon)
3. vulnerability of personnel when operating the equipment
4. weapon effects

An equipment type may have a maximum of eight modes of operation. Furthermore, a given equipment type within a unit may be distributed over several modes of operation at the same time, and the distribution may vary with changing battle conditions during the simulation.

c) Term References:

Equipment type is referenced by retrieving information defining the attributes relating to a specific equipment. Each attribute is associated with a type identification integer, ranging in value from 1 to 80 inclusively. Equipment type is used extensively by the air and ground fire modules. These modules reference equipment type contained within the unit to determine the allocation of weapons against the enemy, the amount of fire to deliver, the resulting damage and casualty, the movement rate of each firing unit, etc. Note that movement rate is computed by equipment type, and is dependent upon the operation of the equipment type. The movement logic (both air and ground) and the detection logic examine various attributes of equipment type. Movement is concerned with the physical limitations of equipment types operating over terrain, and detection is principally concerned with noise generated from vehicles, machinery, etc.; detection capabilities are enhanced when a unit contains equipment like infrared night detection devices (e.g., TVS-4).

3.54.2 Programmatical Definition

a) Term Name:

DBEQ(80,2)	EQWDTH(80)	IEQSIDE(20)	NPTE(80,3)	ROMMX(80)
DIES4KM(80)	GAS4KM(80)	IMAXRE(80,2)	NSTE(80,3)	SLPEMX(80)
DTMTBFI(80)	IAMTE(80,8)	IMINRE(80)	OPE(80)	STATS(40,41,2)
EQCAPAC(80)	IECTH(80)	IPVCE(80,8)	PCPEC(80,8)	UBE(80)
EQNAME(3,80)	IEQCLS(80)	NDXEQ(80,2)	ROFE(80,8)	VCI(80)
EQOVRD(80)	IEQCOD(80)	NDXSTATS(40,2)	ROME(80,8)	

b) Term Description:

All arrays representing attributes of equipment type are listed below: Generally the indices IEQ and JEQ are integers which range from 1 to 80. However, there are instances where the range differs because the arrays contain packed data. Equipment type is largely determined by user input. The majority of the arrays contain data that will remain constant throughout the simulation.

DBEQU(IEQ,J)	Noise level produced by IEQth equipment type: J=1 when unit is not moving. J=2 when unit is moving.
DIES4KM(IEQ)	Amount of diesel fuel in gallons that equipment type IEQ uses to travel one km.
DIMTBFI(IEQ)	The negative inverse of the mean time between failure in minutes for ground equipment type IEQ (1-80)
EQCAPAC(IEQ)	For ground equipments (indicated by IEQCOD(IEQ) > = 0), fuel capacity in gallons of equipment type N, including spare cans carried for aircraft (IEQ COD(IEQ) = -1), fuel capacity of aircraft (pounds)
EQNAME(J,IEQ)	12 character name associated with equipment type IEQ (1-80) J=1 first four characters of equipment name J=2 second four characters of equipment name J=3 third four characters of equipment name
EQOVRD(IEQ)	the maximum diameter in meters tree trunks in a given vegetation class that the IEQth (1 < = IEQ < = 80) equipment type can override
EQWDTH(IEQ)	the width in meters of the IEQth (1 < = IEQ < = 80) equipment type.
GAS4KM)(IEQ)	amount of gas in gallons that equipment type IEQ use to travel one km.

IAMTE(IEQ,IMODE) for ground equipments (indicated by IEQCOD(IEQ) < = 0), ammunition type used by weapon type IEQ in mode of operation IMODE (0 implies no ammunition used).

For aircraft (IEQCOD(IAC) = -1,

IMODE = 1 takeoff delay time for this aircraft (min.)

2 landing delay time for this aircraft (min.)

for air ordnance (IEQCOD(IAC) = -2),

IMODE = 1 number of ammunition type for this equipment, if any

2 number of rounds in a standard load of that ammunition type

3 rate of fire of this equipment (rounds/min)

4 number of drops per pass (applied to bombs)

5 distance between drops (meters)

6 dud probability times 100

7 kill probability times 100 for bridge type

8 kill probability times 100 for bridge type

IECTH(IEQ) Number of pieces of equipment type IEQ which may be lost by a unit before a casualty Alert is generated

IEQCLS(IEQ) For ground equipments (indicated by IEQCOD(IEQ) > = 0), for each ground equipment type IEQ, a general cla-sification for air ordnance, defined as follows:

1=small arms

2=light mortars

3=artillery

4=non-armored vehicles

5=armored vehicles

For air ordnance (IEQCPD(IEQ) = -2),

IEQCLS = 1 IEQ is a 250-lb low-drag bomb

2 IEQ is a 500-lb low-drag bomb

3 IEQ is a 750-lb low-drag bomb

4 IEQ is a 1000-lb low-drag bomb

5 IEQ is a 2000-lb low-drag bomb

6 IEQ is a 250-lb high-drag bomb
7 IEQ is a 500-lb high-drag bomb
8 IEQ is a 750-lb high-drag bomb
9 IEQ is MAVERICK
10 IEQ is SHRIKE
11 IEQ is 2.75 rockets
12 IEQ is 20-mm cannon
13 IEQ is CBU-2
14 IEQ is CBU-24
15 IEQ is ROCKEYE
16 IEQ is NAPALM

Not used for aircraft.

IEQCOD(IEQ) — Equipment category code of given equipment type IEC:

-3-air sensor
-2-air weapon
-1-aircraft
0-not a weapon
1-direct fire weapon
2-indirect fire weapon
3-support fire weapon.
4-airdefense weapon

IEQSIDE(IEQ) — A byte packed array (IEQ=1,20) where each byte I contains a code controlling whether equipment type I will appear on the menus when the red or blue C and C button is pushed.

0=both on red and blue
1=red only
2=blue only
3=neigher red or blue (spare)

IMAXRE(IEQ,ITGTE) — For ground equipments (indicated by IEQCOD(IEQ) > = 0), maximum firing range (in meters) of each weapon type IEQ against primary (ITGTE=1), secondary (ITGTE=2) target elements for air equipment other than an aircraft (IEQCOD(IEQ) = -2 or -3), maximum range at which equipment may may be used (ITGTE = 1)

IMINRE(IEQ)	For ground equipments (indicated by IEQCOD(IEQ) > = 0), minimum firing range (in meters) for weapon (equipment) type IEQ For air equipment other than an aircraft (IEQCOD(IEQ) = -2 or -3), minimum range at which equipment may be used
IPVCE(IEQ,IMODE)	For ground equipments (indicated by IEQCOD(IEQ) > = 0), personnel vulnerability class associated with each mode IMODE of equipment type IEQ. For air equipment other than an aircraft (IEQCOD(IEQ) = -2 or -3), equipment number of the IMODEth allowable aircraft type on which IEQ may be used. (Hence, a maximum of eight aircraft per equipment type.) The first zero encountered implies no other allowable aircraft.
NDXEC(JEQ,JCOLOR)	Index number for array stats for equipment number JEQ(JEX=1,80), color JCOLOR(1=red,2=blue)
NDXSTATS(JEQ,JCOLOR)	Equipment number of the JEXth equipment (JEQ=1,40) in the stats array for color JCOLOR(1=red, 2=blue)
NPTE(IEQ,J)	Equipment numbers of two primary target types for weapon IEQ; -1 implies a personnel target.
NSTE(IEQ,J)	Equipment numbers two secondary target types for weapon IEQ; -1 implies a personnel target.
OPE(IEQ)	Number of men required to operate equipment IEQ in three states : dismounted, mounted, maximum capacity. The three values are packed into the first three bytes of the word respectively.
PCPEC(IEQ)	Expected personnel casualties per equipment casualty for equipment type IEQ
ROFE(IEQ,IMODE)	For ground equipments (indicated by IEQCOD(IEQ) > = 0), rate of fire of weapon type IEQ in each mode

IMODE = 1 fuel expenditure for losing altitude (lb/meter)

2 fuel expenditure for gaining altitude (lb/meter)

3 fuel expenditure at minimum speed, minimum load, best pressure density (lb/min)

4 fuel expenditure at cruise speed (lb/min)

5 fuel expenditure at maximum speed (lb/min)

6 ratio of fuel expenditure rate at maximum load to fuel expenditure rate at minimum load

7 ratio of fuel expenditure rate at worst pressure density to fuel expenditure rate at best pressure density

8 not used

for air ordnance (IEQCOD(IEQ) = -2),

IMODE = 1 fraction of personnel in personnel vulnerability class 1 (standing) and within target area who are killed by this equipment

2 fraction of personnel in personnel vulnerability class 2 (crouching) and within target area who are killed by this equipment

3 fraction of personnel in personnel vulnerability class 3 (prone) and within target area who are killed by this equipment

4 fraction of equipment with IEQCLS = 1 and within target area which is damaged by this equipment

5 fraction of equipment with IEQCLS = 2 and within target area which is damaged by that equipment

6 fraction of equipment with IEQCLS = 3 and within target area which is damaged by this equipment

7 fraction of equipment with IEQCLS = 4 and within target area which is damaged by this equipment

8 fraction of equipment with IEQCLS = 5 and within target area which is damaged by this equipment

ROME(IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0), rate of movement of weapon type I in mode IMODE (unobstructed).

For aircraft (IEQCOD(IEQ) = -1,

IMODE = 1 maximum load aircraft can carry (pounds) at best modeled pressure density (PDBEST)

= 2 maximum altitude of aircraft (meters)

= 3 minimum speed of aircraft (meters/minute)

= 4 cruise speed of aircraft (meters/minute)

= 5 maximum speed of aircraft (meters/minute)

= 6 maximum load aircraft can carry (pounds) at worst modeled pressure density (PDWORST) the correct negative value will insure that an aircraft cannot fly at pressure densities below its capabiltiy

= 7 poorest meteorological visibility in which aircraft can continue its mission (meters)

= 8 not used

For equipment other than aircraft (IEQCOD(IEQ) = -2 or -3),

IMODE = 1 weight of equipment (pounds) including standard ammunition load

= 2 minimum aircraft speed (meters/minute) at which equipment can be used

= 3 maximum speed at which equipment can be used (meters/minute)

= 4 minimum altitude at which equipment can be used (meters)

= 5 maximum altitude at which equipment can be

= 6 for sensors, not used; for ordnance, road crater radius (meters) against road type 1

= 7 for sensors, not used; for ordnance, road crater radius (meters) against road type 2

= 8 for sensors, not used; for ordnance, road crater radius (meters) against road type 3

ROMMX(IEQ) The maximum rate of movement for the IEQth (1 < = IEQ < = 80) equipment under ideal conditions

SLPEMX(IEQ) The maximum slope that the IEQth (1 < = IEQ < = 80) equipment can negotiate

STATS(JEQ,J,JCOLOR)

Equipment and personnel casualty statistics for red killing blue (JCOLOR=1), and blue killing red (JCOLOR=2) equipments. JEQ=1,40 corresponds to NDXSTATS(JEQ,JCOLOR). J=1,41 corresponds to NDXSTATS(J,3-JCOLOR), where J=41 is used for personnel casualties

UBE(IEQ) Weighing factor: importance of equipment type IEQ as a target

VCI(IEQ) Vehicle cone index for the IEQ-th (1 <= IEQ <= 80) equipment

DTMTBFI, EQOVRD, EQWDTH, IEQCLS, ROMMX, SLPEMX, and VCI are data base input variables that are defined in the namelist input deck.

IECTH, NDXEQ, NDXSTATS, and STATS are model generated variables. IECTH is used in subroutine FORMAIN (data statement), NDEXQ and NDXSTATS are used in subroutine INPUT, and STATS is used in subroutines ADW, INIT, OBSDELAY, REDIST, STEP, and WPNFIR.

All other variables are data base input variables that are defined in the equipment input deck.

c) Term References:

The arrays containing data describing equipment type attributes are indexed by equipment type number IEQ(or JEQ). Equipment type number is an integer such that $1 \le$ IEQ,JEX ≤ 80. Arrays indexed by IU indicate that initial values via user input must be specified for these arrays. The index JU indicates that these arrays contain computed

data. Some arrays contain packed data, thus the index will not exactly correspond to equipment type number. However, the data is eventually accessed according to equipment type number and index number. The arrays defining the attributes of equipment type are used by one or more of the following subroutines:

ADJROM	EQIPNR	OTHROMB
ADW	EQUPLINE	PRNTFIR
ADWALRT	FIRAGEN	RADAR
ADWDATA	FIRALO	RAMGEN
AIRABORT	FIREVNT	REDIST
AIRCAS	FIRSORT	RESUPPLY
AIRERROR	FORMAIN	SAVE
AIREVENT	FORRAM	SAVEINP
AIRGRND	FSTOT	SETIMP1
AIRHOTZN	GENFIR	SPTALO
AIRMOV	INIT	STATREP
AMMOINP	INPUT	STEP
AMOVUL	LOWALRT	TGTCAT
AURAL	MANNING	TGTLST
CASREP	MOVMNT	USEFUEL
CBTVAL	NOTGT	WPNEFF
CHGCRT	OBSDELAY	WPNFIR
CRDLIC	ORDPRI	WPVEL
EQINP	ORGFIR	WTSUB

3.55 EVENT NOTICE

3.55.1 English-language Definition

a) Term Name:

Event Notice

b) Term Description:

In CATTS, event notices come from the following sources:

1) Interactive command and control inputs by the instructors via the menus.

2) Events from the pre-scheduled events file.

3) Events input as part of the card run deck (must be the next cards after the namelist portion of the run deck).

Each event notice corresponds to an event due to occur at a particular time. These event notices are maintained in chronological order and released individually at their scheduled times by the mathematical model event processing system (see Section 5.9). For each event notice that is released, the appropriate event processor is activated to modify the appropriate mathematical model variables to implement the desired operation.

c) Term References:

Event notices are processed at the beginning of every simulation minute.

3.55.2 Programmatical Definition

a) Term Name:

INVDT(64)
ITMEVE
IVDT(64)
NEVENT
NEVTP

b) Term Description:

INVDT(64) — Event notice data of the event the math model is currently processing.

ITMEVE — Time (in minutes since simulation start) at which event that is being scheduled will occur (on both pre-scheduled events file and run card deck events).

IVDT(64) — Event notice data of the event that is being scheduled (on both pre-scheduled events file and run card deck events).

NEVENT — The total number of events being input via the card run deck (if there are any events in the run card deck, this variable should be set in the namelist portion of the run deck).

NEVTP — The total number of events being input on the pre-scheduled events deck.

INVDT(64) is a model generated variable that is used in subroutines PREVENT and EVENTS. ITMEVE and IVDT(64) are data base input variables that are defined in the run deck and by card images on the pre-scheduled events file. NEVENT is a data base input variable that is defined in the namelist portion of the run deck. NEVTP is a data base input variable that is defined by a card image at the front of the pre-scheduled events file.

c) Term References:

Both the run deck events and events from the pre-scheduled events file are read-in in subroutine INPUT. Then subroutine PEVENT puts them on the background events file for processing by the events module at the time they are scheduled to occur. The events module is subroutine EVENTS, PEVENT, REVENT, PPEVENT, PREVENT, and EVENTER. The various events are handled by the subroutines as shown below:

Event Type	Name	Subroutine Called to Process the Event
1	Weather	WETHRC
2	Resupply	RESUPPLY

Event Type	Name	Subroutine Called to Process the Event
3	Control measures	CONTMS
4	Split unit (can't be used, no menu)	CRDLIC
5	Generate alert message	ALNUALRT
6	Change unit location	UNLOCATE
7	Unit activation	ACTIVATE
8	Air mission	AIREVENT
9	Ground maneuver	MANEUVER
10	Ground fire	FIREVNT
11	(not used)	
12	(not used)	
13	(not used)	
14	Air defense	AIRDEVNT
15	Preplanned mission	PREPLAN
16	Task organization	TASKORG

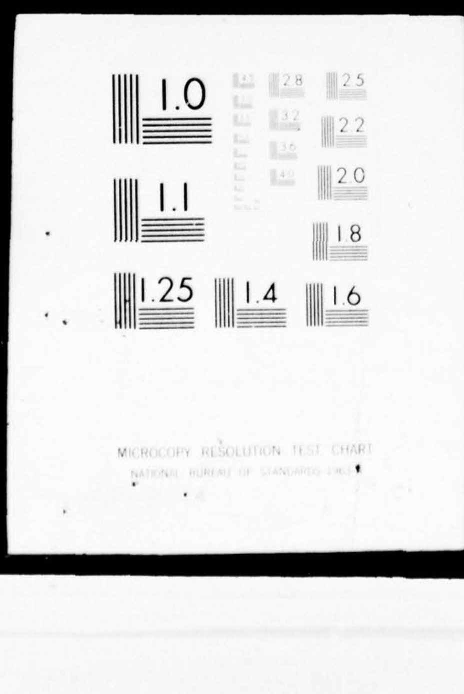
AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

4 OF 8
AD
A038796

4 OF 8

AD
A038796



3.56 FAKE UNIT

3.56.1 English-language Definition

a) Term Name:

Fake Unit

b) Term Description:

Fake units are units having unit numbers ranging from 101 to 150. They provide a representation of higher command units which are implicitly assumed to exist although they are not explicitly represented. Fake units have names but have no other attributes and are not displayed in the area of operation. The following unit numbering convention is established in the CATTS math model:

1) Units 1-100 are legitimate units
2) Units 101-120 are operational groups 1-20
3) Units 121-135 are nonexistent red units
4) Units 136-150 are nonexistent blue units.

c) Term References:

Fake units are used for several purposes in the CATTS math model. They are used principally to simulate higher command units so that a hierarchy of commands is established. This is required when dealing with control measure interactions. Control measures are associated with units, and when a unit crosses a control measure, an alert is generated not only for that unit but for all its subordinate units. A similar requirement exists if operational groupings violate the control measures. Fake units also provide a representation for higher command units which serve as the source from which supplies can be obtained. This provides the instructor with the interactive capability of being able to resupply, at his discretion, any unit with any amount of equipment and/or personnel.

3.56.2 Programmatical Definition

(Not applicable)

3.57 FATIGUE

3.57.1 English-language Definition

a) Term Name:

Fatigue

b) Term Description:

Fatigue is a factor (a fraction between zero and one) which models the effect of human performance degradation. This degradation factor is modeled for dismounted troops and affects overall unit rate of movement. The model assumes that fatigue is experienced uniformly throughout the unit. Thus it suffices to compute the fatigue factor for one typical (as far as height, weight, energy and water consumption, etc.) man in the unit and apply this factor to the total number of dismounted personnel in the unit.

Fatigue is generally considered to be related to the total amount of energy (BTU's) expended by the body. This model assumes that fatigue increases exponentially from 0 to 1 as net energy expenditure over time increases from 0 to 35000 BTU's. The value of 35000 BTU's has been selected as representative of a level of energy expenditure wherein combat personnel have reached a state of virtual exhaustion and thus a performance degradation due to fatigue of 100 percent

c) Term References:

This factor is referenced only by units which have troops traveling on foot (i.e., dismounted). Otherwise, the factor of fatigue does not affect overall unit rate of movement.

3.57.2 Programmatical Definition

a) Term Name:

FATIGU

b) Term Description:

FATIGU is a model generated variable that is used in subroutine ADJROM. It is a fraction having a value between zero and one inclusively. The value depends on the fraction of personnel within the unit moving on foot, and the total amount of energy expended by a typical individual within the unit. This factor is computed during each timestep for dismounted units.

c) Term References:

FATIGU is referenced by the following subroutine:

ADJROM

3.58 FEBA

3.58.1 English-language Definition

a) Term Name:

FEBA

b) Term Description:

In the CATTS simulation of an engagement, a FEBA (forward edge of battle area) is defined for each side, with these FEBAs being parallel reference lines generally indicating the forward edges of forces (Red and Blue units) on each side of an engagement. Each FEBA has a definite location, orientation, and length associated with it. Figure depicts FEBAs for opposing units. The mid-point of a FEBA is called the "location of the FEBA" and is determined by the location of the controlling (i.e., forwardmost) operational grouping, if there is one. If there is no controlling operational grouping, then the unit belonging to the engagement which is nearest to the enemy force has its location taken as the location of the FEBA. From these FEBAs, the engagement axis is drawn as the vector from the Blue FEBA to the Red FEBA. The FEBA length is equal to twice the half-frontage of the respective Blue or Red force (see the term "engagement frontage").

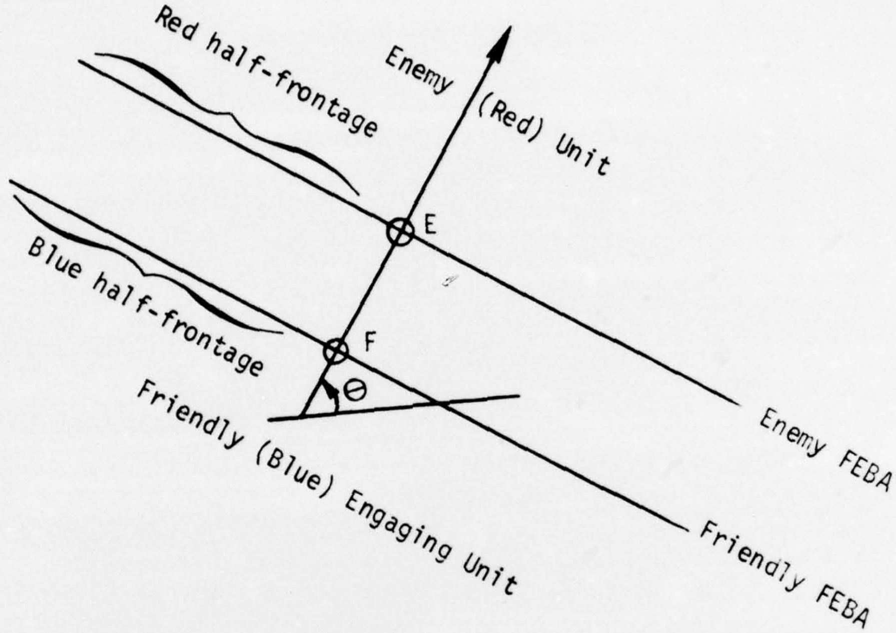
c) Term References:

FEBAs exist only during engagements. A FEBA is defined for each opposing force in the CATTS simulation of an engagement (see Section 5.6, Engagements Module).

3.58.2 Programmatical Definition

a) Term Name:

IFEBAR(JENG,J), IFEBAB(JENG,J)
IBFRNT(JENG), IRFRNT(JENG)
DIREAX(JENG,J)



F = location (midpoint) of friendly FEBA
= (IFEBAR(JENG,1),IFEBAR(JENG,2)) or
(IFEBAB(JENG,1),IFEBAB(JENG,2))

E = location (midpoint) of enemy FEBA
= (IFEBAB(JENG,1),IFEBAB(JENG,2)) or
(IFEBAR(JENG,1),IFEBAR(JENG,2))

↑ = engagement axis

sin θ = DIREAX(JENG,1)
cos θ = DIREAX(JENG,2)
Red half-frontage = IRFRNT
Blue half-frontage = IBFRNT

Figure 3-2. FEBA Definition

b) Term Description:

IFEBAR(JENG,J) is the X (J=1) and Y (J=2) coordinates of center point of Red FEBA (location of controlling Red operational grouping or intersection of Red FEBA with JENGth engagement axis) for 1 ≤ JENG ≤ 12. Similarly, IFEBAB(JENG,J) is the X (J=1) and Y (J=2) coordinates of center point of Blue FEBA in the JENGth engagement for 1 ≤ JENG ≤ 12.

DIREAX (JENG,J)	Is the direction of the JENG-TH engagement axis, Blue to Red. Stored as SIN (J=1), COS (J=2), where 1 ≤ JENG ≤ 12.
IBFRNT (JENG)	Is the half-frontage of Blue force in the JENGth engagement, where 1 ≤ JENG ≤ 12.
IRFRNT (JENG)	Is the half-frontage of Red force in the JENGth engagement, where 1 ≤ JENG ≤ 12.

These variables are model generated variables that are used in the Engagement Module.

c) Term References:

The above variables are referenced in the following subroutines:

ADJDIR	FORRAM
ANYFOE	FRNTGS
CHGCRT	FWDLIN
CHGOPN	LATDST
CLSPTG	NEWENG
DEPLOY	NEWMOV
ENAREA	NGARGN
ENGDIR	OPPLAN
ENGINP	TGTCAT
FINDFO	UN2FEB

3.59 FIRE TYPE

3.59.1 English-language Definition

a) Term Name:

Fire Type

b) Term Description:

There are eight different equipment codes used in the model to categorize the list of eighty equipments. These eight categories are:

1) non-firing ground equipment
2) direct fire weapon
3) indirect fire weapon
4) support fire weapon
5) air defense fire weapon
6) aircraft
7) air ordnance
8) air sensor

Five of the above can be considered fire type: the four types of ground weapons and air ordnance. This code is used throughout the model to determine which equipments to process in a given module. In processing the ground fire equipment (excluding air defense weapons), an indicator is used with which to watch equipment codes. At any given time, this variable indicates one of the following:

1) non-firing ground equipment
2) direct fire weapon
3) indirect fire weapon
4) support fire weapon

Data identifying fire type is input at initialization and not altered thereafter.

c) Term References:

Equipment code is an equipment attribute used throughout the model to distinguish what type of processing an equipment should receive.

3.59.2 Programmatical Definition

a) Term Name:

IEQCOD(IEQ)
KFICTR

b) Term Description:

The array IEQCOD is a data base input variable that is defined in the equipment input deck. It is defined as follows:

IEQCOD (IEQ)	Equipment category code of given equipment type IEQ: -3-air sensor -2-air weapon -1-aircraft 0-not a weapon 1-direct fire weapon 2-indirect fire weapon 3-support fire weapon 4-air defense weapon'

The indicator KFICTR is a model generated variable that is used in subroutines CBTFIR and SPTFIR. It is defined as follows:

KFICTR	A code word indicating which type of fire is being considered: 0 - nonfire 1 - direct fire 2 - indirect fire 3 - support fire.

c) Term References:

IEQCOD(IEQ) is indexed by equipment number IEQ, where IEQ ranges between ($1 \leq IEQ \leq 80$). Equipment code is used to determine which equipments should be processed by a particular module. The following table lists the main subroutine processing each equipment category:

IEQCOD	Category	Subroutine
-3	air sensors	AIRGRND
-2	air ordnance	ADW
-1	aircraft	AIRMOV
0	non-firing ground equip.	NOTGT
1	direct fire	FIRALO
2	indirect fire	FIRALO
3	support fire	SPTALO
4	air defense	AIRCAS

KFICTR is a one word indicator used in the Ground fire module to select the proper equipments to be processed by a particular routine.

3.60 FIXED FRAME OF REFERENCE

3.60.1 English-language Definition

a) Term Name:

Fixed Frame of Reference

b) Term Description:

Locations and directions in the area of operation are determined with respect to a fixed, right-handed, rectangular Cartesian coordinate system. The positive X-axis of this system is oriented in the due east direction (zero degrees) and the positive Y-axis is oriented in the due north direction (ninety degrees).

c) Term References:

All angle and location computations within the area of operation are made with respect to this fixed frame of reference.

3.60.2 Programmatical Definition

(Not applicable)

3.61 FOREGROUND SOFTWARE

3.61.1 English-language Definition

a) Term Name:

Foreground Software

b) Term Description:

The complexity of the controller interactions required in CATTS dictated that large, complex software packages would be required. To increase modularity and speed development, each major interactive function was designed as a separate program. Because the interactive software needs to be interrupt driven, they run as foreground programs, and are often referred to as "foreground software". This means that the interactive (foreground) software is executed at a higher priority than any mathematical model software program residing in the background area. When no requests for processing of an interrupt-driven foreground software program are queued, the background software is executed.

The foreground software is described in detail in Section 4 of the Programming Report.

c) Term References:

(None)

3.61.2 Programmatical Definition

(Not applicable)

3.62 FREEZE

3.62.1 English-language Definition

a) Term Name:

Freeze

b) Term Description:

The principle instructor is provided with the option of halting the CATTS simulation at any time and for any desired length of time. This way, the principle instructor issues a FREEZE command via the command and control subsystem, performs whatever tasks he wants to do while the simulation is halted, and then presses the simulation control switch to continue with the simulation.

c) Term References:

None

3.62.2 Programmatical Definition

(Not applicable)

3.63 FRONT LINE TRACE

3.63.1 English-language Definition

a) Term Name:

Front Line Trace

b) Term Description:

In the CATTS system, front line trace refers to the display of the FEBA's (Forward Edge of Battle Areas) of all currently existing engagements in the mathematical model by the graphic display subsystem. Front line traces are displayed for red and blue engagements.

c) Term References:

Up to 12 engagements involving many units in each engagement are allowed. The front line traces displayed for up to 12 engagements.

3.63.2 Programmatical Definition

a) Term Name:

DIREAX(JENG,J)
IBFRNT(JENG)
IFEBAB(JENG,J)
IFEBAR(JENG,J)
IRFRNT(JENG)
IXPHIB(JENG)
IXPHIR(JENG)
MFIGHT(IUT,ICOLOR)
NGARNG(IUT,ICOLOR)
NNGAGE

b) Term Description:

DIREAX(JENG,J) – Direction of the JENGth engagement axis, blue to red. Stored as sine (J=1) and cosine (J=2). $1 \leq JENG \leq 12$

IBFRNT(JENG) – Half-frontage of blue units in the JENGth engagement. $1 \leq JENG \leq 12$

IFEBAB(JENG,J) — The X (J=1) and Y (J=2) coordinates of the center point of the blue "FEBA" in the JENGth engagement. $1 \leq JENG \leq 12$

IFEBAR(JENG,J) — The X (J=1) and Y (J=2) coordinating of the center point of the red "FEBA" in the JENGth engagement. $1 \leq JENG \leq 12$

IRFRNT(JENG) — Half-frontage of the red units in the JENGth engagement. $1 \leq JENG \leq 12$

IXPHIB(JENG) — X coordinate on blue "FEBA" in engagement JENG rotated by angle. DIREAX in the blue direction of movement.

IXPHIR(JENG) — X coordinate of red "FEBA" in engagement JENG rotated by angle DIREAX in the red direction of movement.

MEIGHT(IUT,ICOLOR)— For each unit type IUT, red or blue, the range in meters from an enemy unit at which a unit must initiate an engagement.

NGARNG(IUT,ICOLOR)— For each unit type IUT, red or blue, the range in meters at which and less than a unit is eligible to initiate an engagement with an enemy unit (depending on each unit's travel code, maneuver control, and current engagement status).

NNGAGE — Number of engagements established (0 to 12).

DIREAX(JENG,J) and NNGAGE are both data base input variables and model generated variables. As data base input variables, DIREAX and NNGAGE are defined in the engagement input deck, and as model generated variables, they are used in the Engagement Module. MFIGHT (IUT, ICOLOR) and NGARNG(IUT,ICOLOR) are data base input variables that are defined in the unit type input deck. All other variables are model generated variables set by the Engagement Module.

c) Term References:

Not all the variables used by the engagement module are listed above. MFIGHT and NGARNG are referenced by unit type and color, and they control the ranges at which the engagement module begins to operate. All the other listed variables are created and used by the engagement module and referenced through the mailboxes by the foreground for display.

3.64 FUEL

3.64.1 English-language Definition

a) Term Name:

Fuel

b) Term Description:

Three types of fuel are represented in the CATTS math model. Gasoline, avaiation and diesel fuel are modeled by specifying for each unit the basic load of each type of fuel required to operate its equipment. Every time step the model computes fuel usage of each unit based on the amount of equipment remaining in the unit and the unit's movement. The usage during the time step is subtracted from the current load. The current load of each type of fuel in the unit is maintained throughout the simulation.

c) Term References:

Amount of fuel is an attribute of units, thus it is referenced by unit number. Note that ground units utilize gasoline and diesel fuel, and air units are concerned only with aviation fuel. The movement logic uses fuel data to determine whether a unit containing fuel-driven equipment has sufficient fuel to continue movement. If not, the unit is forced to halt until it is resupplied. The resupply logic, therefore, also makes reference to fuel. Alerts and status reports indicate when the unit's fuel level is low.

3.64.2 Programmatical Definition

A) Term Name:

BLAVGAS(100),BLGAS(100),BLDIES(100),
CLAVGAS(100),CLGAS(100),CLDIES(100)

b) Term Description:

The floating point arrays BLGAS,BLAVGAS, and BLGAS contain data used to model the basic loads (initial amounts) of the different types of fuel (gasoline, aviation fuel, diesel fuel) required by a unit. This data is user defined by input and remains constant throughout the simulation. The floating point arrays CLGAS, CLAVGAS, and CLGAS record the current load of each of the different types of fuel contained in each unit.

These arrays are updated each time step to reflect the amount of fuel consumption or loss (due to equipment destruction) since the last time step. The current load and the basic load are equal at the beginning of the simulation. Afterwards, the current load is decremented each time step unless the unit is resupplied appropriately.

BLAVGAS(IU) = basic load (initial amount) of aviation fuel, in gallons, for unit IU

BLDIES(IU) = basic load (initial amount) of diesel fuel, in gallons, for unit IU

BLGAS(IU) = basic load (initial amount) of gasoline, in gallons, for unit IU

CLAVGAS(IU) = current load (present amount) of aviation fuel, in gallons, for unit IU

CLDIES(IU) = current load (present amount) of diesel fuel, in gallons, for unit IU

CLGAS(IU) = current load (present amount) of gasoline, in gallons, for unit IU

BLAVGAS and CLAVGAS are model generated variables that are used by the Air Module. All other variables are model generated variables that are used in subroutine USEFUEL.

c) Term References:

The arrays BLAVGAS, BLDIES, BLGAS, CLAVGAS, CLDIES, and CLGAS are all indexed by unit number IU, where IU is an integer ranging from 1 to 100 inclusively. IU refers to ground units when used to reference arrays BLDIES, BLGAS, CLDIES, and CLGAS. IU indexes air units for arrays BLAVGAS and CLAVGAS. The above arrays are used in the following subroutines:

AIREVENT	LOWALRT
AIRMOV	MOVE
CRDLIC	MOVMNT
FORRAM	RESUPPLY
FUELLINE	STATREP
INPUT	USEFUEL

3.65 GRID RESOLUTION

3.65.1 English-language Definition

a) Term Name:

Grid Resolution

b) Term Description:

Grid resolution pertains to relief data, and is defined to be the distance between grid points at which ground elevation data is available. Relief data in the area of operation is organized such that elevation data is given at each of the four corners of a grid square. The length of the side of the grid square is the grid resolution of the relief data.

c) Term References:

The grid resolution is referenced by the Line of Sight/Terrain Sub-module, which deals with modeling the terrain feature of relief, and determining the existence of line of sight between any two points within the area of operation.

3.65.2 Programmatical Definition

a) Term Name:

DELXBS, DELYBS

b) Term Description:

DELXBS - Terrain data grid square X-dimension in meters
DELYBS - Terrain data grid square Y-dimension in meters

These variables are model generated variables that are used in subroutine LOSINP (data statement).

c) Term References:

DELXBS and DELYBS is input by a data statement in subroutine LOSINP. The above FORTRAN variables are referenced by the following subroutines:

LOSINP
LOSVEG

3.66 GROUND-AIR DETECTION

3.66.1 English-language Definition

a) Term Name:

Ground-Air Detection

b) Term Description:

The ground-air detection table is a table of flags indicating whether or not visual ground-air detection has taken place. When an air mission is created, the vector for that air unit is cleared, indicating no detections by ground units. Once the air unit is within visibility range of a particular ground unit, a linear probability calculation is performed and compared against a random number. If favorable, the initial detection flag is set, an alert issued, and the flag remains set until the air unit passes out of visibility range of the ground unit, at which time the flag is reset. The probability of detection consists of the following calculation:

$$1.-(D/V)*(1.-(D/V)^{N}/(1.-D/V)$$

where D = distance between ground and air units (meters

V = maximum visibility (meters)

N = number of samples

see Section 5 for equation justification.

c) Term References:

The ground-air detection table is both a (ground) unit and an air unit attribute. It is used primarily by the air defense function to determine which air units are eligible for air defense fire by a particular ground unit having air defense weapons.

3.66.2 Programmatical Definition

a) Term Name:

IGADET(IU,IAU)

b) Term Description:

IGADET (I,IAU) Bit matrix indicating whether ground unit IGU has already detected air unit IAU. IU is used to determine the first index to array IGADET.

IGADET(I,IAU) is a model generated variable that is used in subroutine GRNDAIR.

This table contains one bit for each ground unit versus each air unit. If that bit is set (=1) the ground unit has detected the air unit; if not set (=0), detection has not occurred. Each time a new detection occurs between a ground unit and an air unit, a Superbee alert is generated.

c) Term References:

IGADET is indexed by both ground and air unit numbers. The air unit number itself, IAU, is used as the second index, $(1 \leq IAU \leq 10)$. The ground unit number is converted into a word/bit position combination. The word number index is determined as follows:

$$I + (IU-1)/32 + 1$$

The bit position is determined as follows, where MOD is the modulo function:

$$\text{bit position} = \text{MOD}(IU,32)$$

IU is the ground unit number in the above expressions, ranging from $(1 \leq IU \leq 100)$. The word number is used as the first index, and the bit position is used to shift to the appropriate bit position within the word. The table is cleared (set to zero) by subroutine CLEARDET for all bit positions relating to a given air unit number when that air unit is created. Subroutine GRNDAIR updates the table each time step. Subroutine AIRCAS uses the status of the table as part of the air defense function.

3.67 GROUND SENSOR

3.67.1 English-language Definition

a) Term Name:

Ground Sensor

b) Term Description:

A ground sensor is any sensor which belongs to a ground unit and is used for detecting other ground units. Ground sensors fall into two basic types: namely, attended and unattended ground sensors. Attended ground sensors are modeled in CATTS as equipment types and include optical devices and ground surveillance radars. Unattended ground sensors are modeled by type as passive fields of sensors and are not included as equipment types. Unattended ground sensor field types include seismic, acoustic, disturbance, and infrared. Their detection ranges are modeled as functions of soil types, moisture, and equipment of the detected unit.

c) Term References:

Ground sensors are referenced by the Target Acquisition Module (see Section 5.3).

3.67.2 Programmatical Definition

a) Term Name:

ISENNT(IGS) for attended ground sensors
IUASFT(IUGS,ICOLOR) for unattended ground sensor fields

b) Term Description:

ISENNT(IGS) is the equipment type number for attended ground sensor type IGS

where:

IGS=1	6 x 30 Binoculars
=2	7 x 30 Binoculars
=3	TVS-4 NOD
=4	TVS-2 Starlight Scope
=5	AN/PPS-5 Ground Surveillance Radar
=6	AN/MPQ-4 Ground Surveillance Radar
=7	AN/TPS-25A Ground Surveillance Radar

IUASFT(IUGS,ICOLOR) is the unattended ground sensor field type number for ground force ICOLOR.

where

$$1 \leq IUGS \leq 10$$

ICOLOR = 1 Red Force
= 2 Blue Force

ISENNT(IGS) and IUASFT(IUGS,ICOLOR) are data base input variables, with ISENNT(IGS) defined in the sensor input deck and IUASFT(IUGS, ICOLOR) defined in the unattended ground sensor input deck.

c) Term References:

ISENNT(IGS) is referenced by subroutines VISUAL and RADAR. IUASFT (IUGS,ICOLOR) is referenced by subroutines UASCK and DALERT.

3.68 GROUND SURVEILLANCE RADAR

3.68.1 English-language Definition

a) Term Name:

Ground Surveillance Radar

b) Term Description:

A ground surveillance radar is a ground sensor used by a ground unit to detect ground targets. Ground surveillance radars are modeled in CATTS as equipment types. Factors considered are:

1) Number and type of ground surveillance radars present in the observer unit
2) Range
3) Terrain concealment
4) Target unit (type, size, mission, and range rate)
5) Observer unit (time in place and suppression)

c) Term References:

Ground surveillance radars are referenced by the Radar Submodule of the Target Acquisition Module (see Section 5.3)

3.68.2 Programmatical Definition

a) Term Name:

ISENNT(IGS)

b) Term Description:

ISENNT(IGS) is a data base input variable that is defined in the sensor input deck. It is the equipment type number for ground surveillance radar type IGS.
where:

IGS=5	AN/PPS-5	Ground Surveillance Radar
=6	AN/MPQ-4A	Ground Surveillance Radar
=7	AN/TPS-25A	Ground Surveillance Radar

c) Term References:

ISENNT(IGS) is referenced by subroutine RADAR.

3.69 HARD TARGET

3.69.1 English-language Definition

a) Term Name:

Hard Target

b) Term Description:

When an air strike mission is created, the controller must select a target type from the menu. If "hard target" is selected, the menu processor associates the designated point with the nearest enemy unit, and that unit's location is used as the target point rather than the point designated. If the target unit moves, the target point is updated. When the ordnance is delivered on the target unit, specific "hard" targets defined at initialization time are attacked to simulate the delivery of guided weapons or highly accurate unguided weapons against specific ground equipments. A table of kill probabilities is used to evaluate the effect of each ordnance delivery. The area target logic is then used to account for casualties in the vicinity of the impact.

c) Term References:

Hard target is a specific designation for an air unit on a strike mission and is set at the time the air unit is created, and used when ordnance is delivered.

3.69.2 Programmatical Definition

a) Term Name:

NTGTYP(IAU) = unit number

b) Term Description:

The array NTGTYP(IAU), defined as

NTGTYP(IAU)	Type of target for air unit IAU = ground unit number or code for road, bridge, or area target.

is an interactively generated variable that is created from the air menu. It is set to the unit number under attack when an air strike mission is implemented.

c) Term References:

The index to NTGTYP(IAU) is the air unit number, where ($1 \leq IAU \leq 10$). Subroutine AIREVENT sets NTGTYP(IAU) to the unit number designated in the event notice for hard target, and ADW delivers the ordnance against specific equipments in the ground units.

3.70 IMPACTING FIRES

3.70.1 English-language Definition

a) Term Name:

Impacting Fires

b) Term Description:

Impacting fires symbols are generated at the point of impact of all support fire (equipment code = 3) and air ordnance. A symbol is drawn of the color of the fires. For ground support fire directed at enemy units, the symbol is located at the unit's center of mass. A flag is set on the unit and the graphics display programs draws a symbol on the unit. For ground support fire directed at X,Y points, bridges, and roads, the impacting fires symbol is displayed at the point designated by the fire command. The point is saved from the time the fire command is created, and placed in an array for the graphics display programs to check. For air strikes, the impacting fires symbol is displayed at the calculated point of impact of the air ordnance against all types of targets. This calculated point is placed in the same array as impact points from ground support fire. A special array of pointers is used to distinguish air impact points in order to draw an additional special air symbol over the point. The unit flags, impacting fire location array, and special air impact pointer array are set each minute as fires occur, first for air ordnance delivery in the Air module, then for ground support fire delivery in the Ground fire module. At the conclusion of the time step, all unit flags and impacting fires arrays will have been copied into graphics display program areas for display the next time step. Impacting fires symbols, therefore, always indicate fires that occurred during the last time step. All impacting fires flags and arrays will have been cleared by the beginning of the next time step after the fires occurred.

c) Term References:

The firing function of the Air Module, and the ground fire module set the impacting fires arrays for the purposes of graphic display.

3.70.2 Programmatical Definition

a) Term Name:

IXYIM(I),IXYIMPTR(I),IUNIM(I),IUNIMXY(I),IMPORO(I)

b) Term Description:

The following arrays are model generated variables that are used by the Air and Ground Fire Modules. They contain impacting fires data:

IUNIM(I)	Bit-packed array indicating which units were fired on during the minute (1=fired on), where word 1, bit 0 is the indicator for Unit 1, etc.
IUNIMXY (I)	Four word array of bit flags indicating if a unit has been hit with fire directed at an XY point; such units are then flagged not to have an impacting fire symbol at their center, since the XY-point symbol is within the area of the unit
IXYIM (I)	Each element of the array, if non-zero, indicates the XY location of an impacting fire, as follows: - sign of the word indicates red(-) or blue (+) - after converting to a positive number, bits 0-15 -- X location bits 16-31 -- Y location
IXYIMPTR(I)	Pointer into IXYIM array used by air units deliverling ordnance, enabling a special symbol to be drawn for air imnacting fires
IMPORD (I)	Four word array of bit flags indicating if fire at an XY point just began, in which case an alert is sent; bit positions correspond with impacting fires array IXYIM

See the next section for a complete discussion of the impacting fires tables. The arrays IXYIM and IXYIMPTR are set for air-delivered ordnance in subroutine ADW. For ground support fire, subroutines SETIMP1 and SETIMP2 are called by SPTALO to set array IXYIM. SETIMP1 and SETIMP2 also flag units that are being affected by X,Y point,

bridge or road fire by setting the appropriate bit in array IUNIMXY. Subroutine STEP sets array IUNIM to indicate all units being fired on directly by ground support fire. Those flagged in IUNIMXY are filtered out by COMPLEMENT-ing then AND-ing, so that no more than one impacting fire symbol will be displayed in a given unit's area.

c) Term References:

Array IXYIM(I) has 100 entries for both ground and air impacting fires symbols, so that ($1 \leq I \leq 100$). A maximum of 10 of those entries can be used for air, and are distinguished by array IXYIMPTR (IAU) where, ($1 \leq IAU \leq 10$). JXYIMPTR allows one entry per air unit (maximum of 10 air units allowed in CATTS), and contains the index into IXYIM for air unit IAU. Arrays IUNIMIUNIMXY, and IMPORD are 4-word bit packed arrays.

The ground unit number is converted into a word/bit position combination. The word number index is determined as follows:

$$I = (IU-1)/32 + 1$$

The bit position is determined as follows, where MOD is the modulo function:

$$\text{bit position} = \text{MOD}\ (IU,32)$$

IU is the ground unit number in the above expre-sions, ranging from ($1 \leq IU \leq 100$). The word numbers is used as the first index, and the bit position is used to shift to the appropriate bit position within the word.

The following subroutines are the principal users of the impacting fires arrays.

ADW
SETIMP1
SETIMP2
STEP

See the discussion in the table section for impacting fires table for a complete discussion of the use of these arrays.

3.71 INPUT EQUIPMENT MOVEMENT RATE

3.71.1 English-language Definition

a) Term Name:

Input Equipment Movement Rate

b) Term Description:

Every ground equipment has up to a maximum of eight modes of operation. There is a desired speed at which the equipment type can move when operating in a particular mode. This speed is assumed to be the ideal speed, unencumbered by terrain or suppression factors, while operating in a certain mode. Thus for every ground equipment type, as many as eight speeds associated with each of up to eight modes of operation, must be user defined via input.

c) Term References:

Equipment movement rate is an attribute of equipment type, hence it is referenced by equipment type number (an integer between 1 and 80 inclusively). This movement rate is used to compute the average movement rate of a given equipment type within a unit. The average rate is weighted appropriately by the fractions of the equipment type operating over each of its several modes. Average movement rate for a particular equipment type is conducted by the ground fire module.

3.71.2 Programmatical Definition

a) Term Name:

ROME(80,8)

b) Term Description:

The floating point array ROME is a data base input variable that is defined in the equipment input deck. It contains the desired movement rate (in meters/minute) of all ground equipment types operating in each of their modes of operation.

ROME(IEQ,IMODE) = the desired movement speed in meters per minute of ground equipment type IEQ($1 \leq$ IEQ ≤ 80) when operating in mode IMODE ($1 \leq$ IMODE ≤ 8)

c) **Term References:**

The array ROME(IEQ,IMODE) is indexed by equipment type IEQ, where IEQ is an integer between 1 and 80 inclusively, and modes of operation IMODE where IMODE is an integer between 1 and 8 inclusively. The array ROME is used in the following subroutines:

AMOVUL
EQUINP
ORGFIR

3.72 INTERDICTION

3.72.1 English-language Definition

a) Term Name:

Interdiction

b) Term Description:

One support fire mode is designated for interdictory fires (mode 5). This mode represents fire at enemy units that are near and might become involved in an engagement. Such units are identified by their locations within a prescribed rectangular region surrounding each close-support-fire-region. Figure 3-3 illustrates the interdictory-fire region of an engagement. If such a region is defined relative to the friendly FEBA, it consists of subregions C and D in the figure; if it is defined relative to the enemy FEBA, it consists of subregions D and E in the figure. The dimensions of an interdictory-fire region, as shown in the figure, correspond to input values entered as part of the fire support inputs. Any enemy unit lying within one of these regions is classed as an interdictory-fire target.

c) Term References:

Interdictory area values are input at initialization for each color. They are used exclusively in the support fire allocation function.

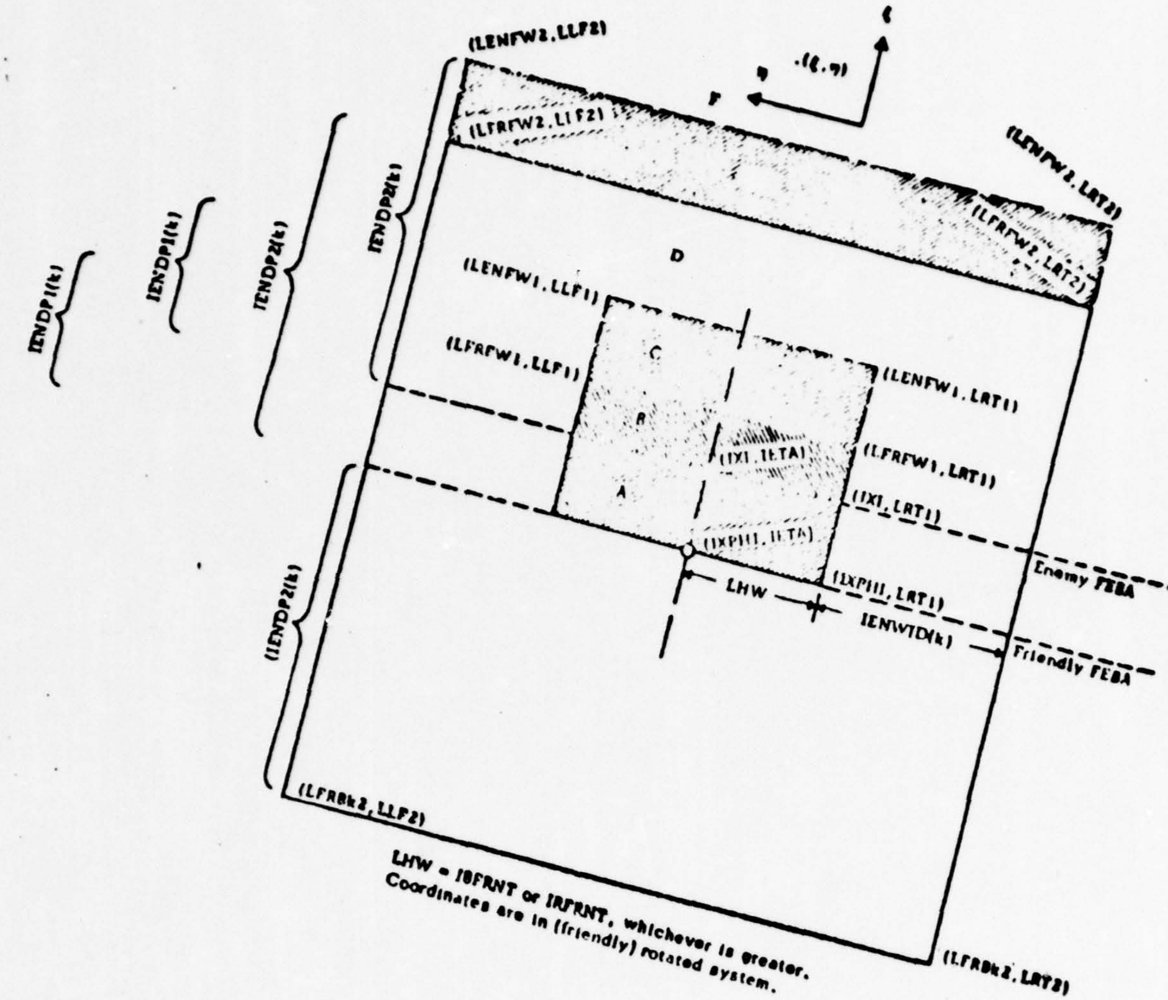
3.72.2 Programmatical Definition

a) Term Name:

IENDR2(I COLOR),IENWID(ICOLOR)

b) Term Description:

IENDP2 (ICOLOR)	In an engagement, outer region in which interdictory fire targets may be assigned. This depth is measured either from friendly FEBA or enemy FEBA, depending upon value of fire support weapon code. ICOLOR = 1 red firing against blue ICOLOR = 2 blue firing against red.



Note: Nature of target (close support or interdiction) is determined relative both to friendly FEBA (for U = 1 in IISTAB) and to enemy FEBA (for U = 2).

Figure 3-3. Close Support and Interdiction Target Areas

IENWID (ICOLOR) Lateral distance beyond the half-width (IRFRNT or IBFRNT) of an engagement in which interdictory fire targets may be assigned. Defines the lateral limits of the region in which fire support of this type is carried out. Lateral limits for close-support fire are IRFRNT or IBFRNT, whichever is greater (to each side of FEBA location):

ICOLOR = 1 Extension for red firing on blue

ICOLOR = 2 Extension for blue firing on red.

These variables are data base input variables that are defined in the fire support input deck.

c) Term References:

IENDP1(COLOR) and IENWID(ICOLOR) for I = 1,2 are input values defined at initialization time. ICOLOR is a flag indicating whether the values pertains to RED or BLUE. The following subroutines use one or more of the above variables:

CLSPTG

FSPINP

3.73 LINE OF SIGHT

3.73.1 English-language Definition

a) Term Name:

Line of Sight

b) Term Description:

Line of sight refers to the ability of a given unit to see an enemy unit, taking into account the terrain features of relief and vegetation Both the observer unit and the target unit are represented by points at the center of each unit. For modeling and computational purposes, each unit is assumed to have a characteristic height (which is a function of unit type) located at its center point.

The line of sight determination is conducted in two steps. The first step considers intervening relief and yields one of two verdicts: obstruction by relief, which causes the line of sight calculations to cease immediately for that unit, or no obstruction, which leads to step two, a detailed examination of possible obscuration by vegetation features. When the line of sight is obstructed, no part of the enemy unit is visible to the observing unit.

Vegetation considerations provide a probabilistic verdict which gives rise to the expected fraction of the enemy unit exposed to the observing unit. The probabilistic verdict reflects the fact that intervening regions of dense vegetation will yield a small fraction of enemy unit exposure, whereas regions devoid of vegetation provide a large amount of exposure. The fraction of exposure is converted to an integral percent number. Thus, the line of sight for a given unit can be spoken of in terms of percent of exposure, where zero percent means that the enemy unit it attempts to see is entirely concealed by relief and/or vegetation.

c) Term References:

Line of sight is referenced by the detection and firing logic in the CATTS math model. Before a unit can allocate direct fire weapons upon another unit, detection via line of sight must have been established. The line of sight verdicts are referenced by unit. For a given unit, a percent of exposure is computed for every eligible enemy unit.

3.73.2 Programmatical Definition

a) Term Name:

LOSPROB(100,25)

b) Term Description:

LOSPROB is a model generated variable that is used in the main program CMAIN, and subroutines CRDLIC, PLOSINP, INPUT, and LOSCOMP. It is a byte packed array containing the percent of unit K exposed to unit I, where K is the Kth byte from LOSPROB(I, 1). The range of exposure is zero for total concealment and one-hundred for totally exposed. The line of sight verdict is not necessarily updated every time step, since line of sight computation is time consuming. For a given unit, the line of sight verdicts are recomputed depending on the following factors:

1) the observer/target units are within a threshold range
2) the operational states of the units involved
3) the types of units involved
4) whether the units involved have changed then positions significantly since the last time step.

c) Term References:

The line of sight verdict is referenced by observing Unit (I) and target unit (K), where the percent of unit K exposed to Unit I is determined by the integer stored in the Kth byte from the byte address of LOSPROB(I,1). The array LOSPROB is used in the following subroutines:

CMAIN	INPUT
CRDLIC	LOSCOMP
DETECT	RADAR
DLOCINP	STATREP

3.74 MAFIA

3.74.1 English-language Definition

a) Term Name:

Mafia

b) Term Description:

MAFIA (Maneuver and Fire Analyzer) is a deterministic time step simulation of a combined arms battle, programmed entirely in FORTRAN IV. It was designed to analyze in detail the close combat arena between opposing forces of brigade size or smaller in relatively short duration battles ranging from several minutes to several hours. The model simulates combat operations with the objective of obtaining realistic results in terms of combat effectiveness indicators and of varying the inputs to determine the effects on the indicators. MAFIA is not interactive. All entities and decision rules are pre-defined, and the model automatically conducts the battle from the point of initialization, without interruption, through to its completion. MAFIA was the starting point of the CATTS math model development.

c) Term References:

None

3.74.2 Programmatical Definition

(Not applicable)

3.75 MAILBOX

3.75.1 English-language Definition

a) Term Name:

Mailbox

b) Term Description:

The mailbox is a common data storage area for all the various programs. The addresses of the math model variables needed by the foreground programs are stored here. Also stored here are a number of flags used for communications between programs - for example, Command and Control sets a flag in the mailbox to inhibit Graphic Display while a menu is being used, to avoid having the menu display erased or overwritten.

c) Term References:

The math model uses the mailbox area to store printers to data needed by the foreground programs for display, command control, or other foreground purpose. These addresses are loaded at systems initialization and not reloaded thereafter.

3.75.2 Programmatical Definition

a) Term Name:

IPT(I)--local array to subroutine FORRAM

b) Term Description:

IPT(I) is a model generated variable that is used in subroutine FORRAM (sets up words 38 through 149, with the remaining words not being used or being set up by the foreground software).

The mailbox contains the following data:

WORD	USE
0	Time step flag
1	Bottom of screen for controller 1
2	Bottom of screen for controller 2
3	Bottom of screen for controller 3

Used by MAP/VIDEO Subsystem	4	X-Bore sight coordinate for controller 1
	5	Y-Bore sight coordinate for controller 1
	6	X-Scale factor for controller 1
	7	Y-Scale factor for controller 1
	8	X-Coordinate of optical center for controller 1
	9	Y-Coordinate of optical center for controller 1
	10	Address of TCXERR
	11	Address of ICYERR
	12	Address of ICXS
	13	Address of ICYS
	14	Address of ICXI
	15	X-Bore sight coordinate for controller 2
	16	Y-Bore sight coordinate for controller 2
Used by MAP/VIDEO Subsystem	17	X-Scale factor for controller 2
	18	Y-Scale factor for controller 2
	19	X-Coordinate of optical center for controller 2
	20	Y-Coordinate of optical center for controller 2
	21	Address of ICYI
	22	Camera motion inhibit flag for controller 1
	23	Camera motion inhibit flag for controller 2
	24	Camera motion inhibit flag for controller 3
	25	NOT USED
	26	X-Bore sight coordinate for controller 3
	27	Y-Bore sight coordinate for controller 3
	28	X-Scale factor for controller 3
	29	Y-Scale factor for controller 3
	30	X-coordinate of optical center for controller 3
	31	Y-coordinate of optical center for controller 3
	32	NOT USED
Used for Simulation Control	33	Simulation control type
	34	Variable information
	35	Days
	36	Hours
	37	Minutes

WORD	USE
38	IONROAD*
39	NOT USED
40	NUMIT*
41	NRU*
42	ITPPL*
43	IXY*
44	EQNAME*
45	ITEQU*
46	TOTEQU*
47	NETU*
48	IUWID*
49	IUDEP*
50	PDIR (1,1) *
51	PDIR (1,2
53	UNNAME *
54	BROMU *
55	ITIME *
56	PERS*
57	PERSI*
58	EQINIT*
59	IEQCOD*
60	NTAMU*
61	NETAMU*
62	NETAMX*
63	BLGAS*
64	BLDIES*
65	CLGAS*
66	CLDIES*
67	IRATT*
68	IWEATHER*
69	IUNIT*
70	ITYPEU*
71	NOT USED
72	NOCMT*

* Indicates pointer to variable in the mathematical model data base

WORD	USE
73	ICM*
74	NPPM*
75	IPEPLAN*
76	IWCLS*
77	NDAYE*
78	LPLTN*
79	LCOMPNY*
80	LBATLN*
81	LALL*
82	NOPLTN*
83	NOCOMPNY*
84	NOBATLN*
85	NOALL*
86	LISTUN*
87	ISTATU*
88	ISENNT*
89	IEQCOD*
90	NOUAS*
91	UASFRS*
92	UASFX*
93	UASFY*
94	UASNAME*
95	IUASFT*
96	UASTRS*
97	IMAXRE*
98	IOBSTYPE*
99	IOBRB*
100	IOBX*
101	IOBY*
102	NSEGMT*
103	IMDEADCO*
104	MENIN*
105	NAMOU*
106	ISTOP*
107	MNRSUT*

* Indicates pointer to variable in the mathematical model data base

WORD	USE
108	SIGU*
109	SIGRADMX*
110	NFDU*
111	NLDU*
112	ITRAN*
113	TIMXY*
114	IXAIR*
115	IYAIR*
116	NAIRPTS*
117	IOGSTAT*
118	NEQUIP*
119	NUNAMO*
120	IAMMONAM*
121	BLAYGAS*
122	CLAVGAS
123	UELEV*
124	MENNOW*
125	ISENNAME*
126	IEQSIDE*
127	IOBWIDTH*
128	DIREAX*
129	IFEBAR*
130	IFEBAB*
131	IXPHIR*
132	IXPHIB*
133	NNGAGE*
134	IRFRNT*
135	IBFRNT*
136	INOG*
137	IOGCDS*
138	IUNIM*
139	NIMFIRES*
140	IXYIM*

* Indicates pointer to variable in the mathematical model data base

WORD	USE
141	IZAIR*
142	IOPSTU*
143	IAMOSIDE*
144	NUMCNTRL*
145	ITMSTRT*
146	LACT*
147	IYXIMPTR*
148	NO. Of loops to perform for audio display*
149	ITESTSC*
150	NOT USED
.	.
.	.
.	.
197	NOT USED
198	Ramtek handler cursor inhibit flag
199	Location of cursor instructor buffer

* Indicates pointer to variable in the mathematical model data base

c) Term References:

At system initialization, subroutine FORMAIN call subroutine FORRAM, which stores the appropriate addresses in local array IPT, which is dimensioned 112. FORRAM is not called thereafter during the exercise.

3.76 MAINTENANCE ATTRITION

3.76.1 English-language Definition

a) Term Name:

Maintenance Attrition

b) Term Description:

The maintenance attrition model implemented in CATTS uses an exponential decay model based on the mean-time-between-failures (MTBF), expressed in minutes. It is used to simulate the realistic maintenance attrition of ground equipment.

c) Term References:

Maintenance attrition factors are attributes of ground equipment and are stored in an array by equipment type.

3.76.2 Programmatical Definition

a) Term Name:

DTMTBFI(ITEQU(I,J))

b) Term Description:

Maintenance attrition factors are uniquely defined by the data stored in the array DTMTBFI(ITEQU(I,J)). These factors are data base input variables that are defined in the namelist input deck. They are expressed as the negative inverse of the MTBF for each ground equipment type. The default values for DTMTBFI are all set at $-1.*10^{-6}$, which corresponds to a MTBF of one million minutes.

c) Term References:

The array describing the maintenance attrition factors is indexed by ITEQU(I,J), which is a list of (J=1 to 14) equipment types in Unit I. This array appears in subroutine STEP. The default values, set at $-1.*10^{-6}$, are in routine FORMAIN.

3.77 MANEUVER CONTROL

3.77.1 English-language Definition

a) Term Name:

Maneuver Control

b) Term Description:

Maneuver control pertains to the interactive capability of positioning, moving, and specifying certain operational states for individual units. Maneuver control also includes the capabilities of dictating the unit's intact (i.e., seek, engage as necessary, or avoid enemy contact) while traveling, as well as specifying whether the unit can move in a mounted or dismounted mode. Furthermore, the type of movement that can be chosen is related to the immediate destination of the unit:

1. the unit may move towards another unit
2. the unit may move along a pre-defined route (i.e., control measure), or a newly generated route
3. the unit may to a specific point

Maneuver control can be applied to a unit immediately, or at some time in the future by defining a maneuver control event.

c) Term References:

Maneuver control is referenced by the command and control sybsystem when a maneuver control event is created and written for processing on the foreground events file.

3.77.2 Programmatical Definition

(Not applicable)

3.78 MANNING EQUIPMENT

3.78.1 English-language Definition

a) Term Name:

Manning Equipment

b) Term Description:

Personnel in a unit are distributed over the total number of pieces of equipment in the unit, causing certain equipment to be manned. Unmanned equipment does not function actively, but can be destroyed by enemy fire. Equipment manning is done on a priority basis, where the equipment priorities are basically a function of unit type (see discussion of subroutine REDIST in Section 5.0). Manned equipment is used primarily to determine a unit's ability to detect and fire at the enemy, and to calculate a movement rate for the unit. Data is input at initialization time defining three manning levels for each equipment: dismounted, mounted, maximum vehicle carrying capacity. This data is not changed after initialization.

c) Term References:

Manned equipment is a unit attribute used by most major functions, including manning distribution, detection, fire, and movement. The number of people required to man each equipment is an equipment attribute used by the manning distribution function.

3.78.2 Programmatical Definition

a) Term Name:

USEEQU(IU,J)

b) Term Description:

USEEQU (IU,J)	Number of pieces manned for the Jth equipment type carried by Unit I.
OPE (IEQ)	Number of men required to operate equipment IEQ in three states: dismounted, mounted, maximum capacity. The three values are packed into the first three bytes of the word respectively.

Once each time step, personnel are distributed over the various equipments operating in the unit according to a list of equipment priorities associated with the unit's type. The result of this distribution is represented in the USEEQU array for each equipment in the unit. The USEEQU array values are model generated variables that are used in subroutines ADW, AIREVENT, AIRHOTZN, AIRMOV2, INIT, REDIST, RESUPPLY and STEP. Array OPE is set via unit input definition and never changed.

c) Term References:

USEEQU(IU,J) is indexed by unit number, IU, where ($1 \leq IU \leq 100$); index J is a dummy index in the range ($1 \leq J \leq 14$) for up to 14 different equipments in a unit. A large number of subroutines use USEEQU. Primarily, subroutine REDIST sets the values for USEEQU early in each time step, and some major detection, firing, and movement subroutines use the results, among which are:

AURAL
EFFNS
FIRALO
ORGPRI
RADAR
SPTALO
VISUAL
WPNEFF
WPNFIR

Subroutine MANNING determines the appropriate manning level to use for each equipment as a function of a unit's equipment number as an index, ranging from ($1 \leq IEQ \leq 80$) containing up to three byte-size values for three equipment manning levels.

3.79 MATH MODEL

3.79.1 English-language Definition

a) Term Name:

Math Model

b) Term Description:

The term math model is synonymous with background software in CATTS, since the entire model resides alone in the background area. It is composed of the functional modules discussed in Section 5. The term math model is descriptive of the CATTS background software because it models a combined arms combat situation through the use of programming entities and attributes, which in actuality are a series of numbers in the computer, not actual hardware deployed ina field exercise involving real people. The model is mathematical in nature in that it uses somewhat sophisticated mathematical formulations to determine outcomes of various events occurring in the model, such as the probability of detection among opposing units, the weapons effects casualty computations, and many others.

c) Term References:

None

3.79.2 Programmatical Definition

(Not Applicable)

3.80 MAXIMUM EQUIPMENT MOVEMENT RATE

3.80.1 English-language Definition

a) Term Name:

Maximum Equipment Movement Rate

b) Term Description:

The maximum equipment movement rate is the absolute top speed a given equipment type can attain under ideal conditions. This maximum speed is idependent of the modes of operation defined for the equipment type.

c) Term References:

The maximum speed of an equipment is an attribute of equipment type, hence it is referenced by equipment type number (an integer between 1 and 80 inclusively). The maximum speed, when degraded by terrain factors, gives an effective upper bound to what rate of speed the equipment is allowed to have. This upper bound is established and used by the ground fire module to compute the rate of movement for the equipment type.

3.80.2 Programmatical Definition

a) Term Name:

ROMMX(80)

b) Term Description:

The floating point array contains the top speed (in meters/minute) of all ground equipment types.

ROMMX(IEQ) = the maximum rate of movement for the IEQth ($1 \leq$ IEQ ≤ 80) equipment type traveling under ideal conditions.

ROMMX is a data base input variable that is defined in the namelist input deck. It remains constant throughout the simulation. Top speeds of vehicular equipment can usually be referenced by manufacturer's handbooks.

c) **Term References:**

The array ROMMX(IEQ) is indexed by equipment type IEQ, where the integer subscript **IEQ** ranges from 1 to 80 inclusively. ROMMX is referenced by the following subroutines:

AMOVUL
ORGFIR

3.81 MAXIMUM RANGE

3.81.1 English-language Definition

a) Term Name:

Maximum Range

b) Term Description:

Maximum range is used for ground fire weapons to determine if a target is within primary or secondary range of the weapon. Each ground fire weapon has a set of primary and secondary target elements defined as part of the data base inputs. For fire commands, only primary range is checked to determine if the target unit is within range. For automatic fire allocation, the primary max range is checked for target units having primary target elements only. The secondary max range is then checked for target units having secondary target elements if the allocation could not be done on the primary max range and set of target elements.

Maximum range for air sensors is used to determine if a ground object is within detection range of an air unit equipped with a sensing device.

c) Term References:

Primary and secondary maximum ranges are equipment attributes used primarily in ground fire allocation, and also in air-ground detection.

3.81.2 Programmatical Definition

a) Term Name:

IMAXRE(IEQ,J)

b) Term Description:

IMAXRE (IEQ,J)	Maximum firing range of each weapon type IEQ against primary (J=1), secondary (J=2) target elements, where the range is in units of meters; maximum detection range of each air and ground sensor in meters (J=1), not used for (J=2).

IMAXRE(IEQ,J) is a data base input variable that is defined in the equipment input deck.

c) Term References:

IMAXRE(IEQ,J) is referenced by equipment member, where ($1 \leq IEQ \leq 80$). For ground firing weapons ($1 \leq IEQ \leq 21$ or $26 \leq IEQ \leq 50$ yields ground equipments <u>and</u> IEQCOD(IEQ) = 1,2, or 3 indicates firing weapons), IMAXRE(IEQ,1) is the primary range, IMAXRE(IEQ,2) is the secondary range, of the weapon. For air sensors ($51 \leq IEQ \leq 55$), IMAXRE(IEQ,J) contains a maximum detection range for J=1 (not used for J=2). The array is not used for other values of IEQ. The following subroutines use IMAXRE:

AIRCAS	FIRSORT
AIRGRND	FORRAM
AIRHOTZN	GENFIR
CBTVAL	RADAR
EQINP	SPTALO
FIRALO	WTSUB

3.82 MENU

3.82.1 English-language Definition

a) Term Name:

Menu

b) Term Description:

Menus in the CATTS system are used to dynamically display information used during the command and control process. In initiating a command and control sequence, a controller first selects a command and control function by depressing the appropriate switch on his control panel. At this point a display (menu) appears on the lower 1/3 portion of the controller's t.v. monitor. With the aid of his graph pen, the controller selects one of several options from the menu. Each selection causes a corresponding change in the menu indicating the selection made. At the end of a command and control sequence, the menu disappears from the controller's t.v. monitor.

c) Term References:

None

3.82.2 Programmatical Definition

(Not applicable)

3.83 MINEFIELD

3.83.1 English-language Definition

a) Term Name:

Minefield

b) Term Description:

A minefield in the CATTS math model is represented by a rectangle defined by a line segment and a width. The line segment is given by the X-Y coordinates of its endpoints, and the width is given by a positive integer number. This line segment, when extended infinitely in both directions is referred to as the center line. With this data, the software automatically generates the X-Y coordinates of the four corners of the rectangle representing the minefield.

Gaps between a minefield can be modeled. A minefield is allowed to have at most two gaps. To simulate gaps, the line segment defining the minefield is separated into partitions: two partitions model one gap, three partitions model two gaps. The partitions are subsegments which are disjointed from one another and must lie along the same center line. The X-Y coordinates of the endpoints of each partition must be specified. In other words, if one gap is modeled, two partitions must be formed, thus two sets of X-Y coordinates (i.e., endpoints) must be defined. Similarly, two gaps require three sets of endpoints.

The software will generate the corners of each rectangle corresponding to each partition. In effect, the modeling of gaps makes the minefield appear to be a set of disjoint rectangles, called sections, having a commong center line.

Currently, a maximum of twenty minefields may exist simultaneously in the model. They must be pre-defined via input.

c) Term References:

Minefield is a type of obstacle used by the movement logic to impede the progress of units and operational groupings.

3.83.2 Programmatical Definition

a) Term Name:

MINEDATA(20),MNEFLDXY(4,3,20),NFIELD

b) Term Description:

The data describing minefields is computed from obstacle input data. Input for a given minefield is a width and a set of X-Y coordinates designating the endpoints of line segments lying along the center line of the minefield. The number of these line segments is input also. Subroutine MINEFLDS examines the obstacle type array IOBSTYPE, searching only for obstacles of type three - minefields. For each obstacle of type three, data generated to simulate the minefield. Note that out of fifty obstacles that may exist concurrently, a maximum of twenty may be minefields. The following FORTRAN variables and arrays define the set of minefields fully:

NFIELD = Total number of minefields in the model; this number can not exceed twenty.

MINEDATA(JMNFLD) = Halfword packed array containing data pertaining to the JMNFLD-th (1< = JMNFLD < = 20) minefield; note that there can be at most 20 minefields in the model; the first halfword contains the obstacle number (an integer between 1 and 50 inclusively) of the JMNFLD-th minefield and the second halfword contains the number of sections (integer between 1 and 3 inclusively) making up the JMNFLD-th minefield.

MNEFLDXY(I,J,JMNFLD) =

Halfword packed array containing the X and Y coords. of the endpoints of the line segments describing the JMNFLD-th (1 < = JMNFLD < = 20) minefield; specifically the first halfword contains the X coord./4 of the endpoint in the Ith (1 < = < I < = 4) segment of the Jth (1 < = < J < = 3) section of the JMNFLD-th (1 < = JMNFLD < = 20) minefield; similarly the second halfword contains the Y coord./4 of the endpoint in the Ith segment of the Jth section of the JMNFLDth minefield.

MINEDATA, MNEFLDXY, and NFIELD are model generated variables that are used in subroutine MINEFLDS (subroutine MINEFLDS uses data input in the mine obstacle fortification deck).

c) Term References:

The data described in the above arrays is referenced by the minefield number, JMNFLD, where JMNFLD is an integer ranging from one to twenty inclusively. The minefield variables and arrays are used in the following subroutines:

BRCHPATH
MINEFLDS
OBSTACLE
OBSWIDTH

3.84 MINIMUM RANGE

3.84.1 English-language Definition

a) Term Name:

Minimum Range

b) Term Description:

Minimum range is used for ground fire weapons to determine if a target is far enough away for fire to be effective. It is especially meaningful for mortars, artillery and other launched-projectile weapons. Minimum range is also used for air sensors to determine if a ground object is far enough away from the air unit for use of the equipment.

c) Term References:

Minimum range is an equipment attributes used primarily in ground fire allocation, and also in air-ground detection.

3.84.2 Programmatical Definition

a) Term Name:

IMINRE(IEQ)

b) Term Description:

IMINRE (IEQ)	Minimum firing range for ground weapon (equipment) type IEQ below which the weapon will not fire. Minimum detection range for air sensor (equipment type N) below which the sensor will not detect.

IMINRE(IEQ) is a data base input variable that is defined in the equipment input deck.

c) Term References:

IMINRE(IEQ) is referenced by equipment number, where $1 \leq IEQ \leq 80$. For ground equipment numbers ($1 \leq IEQ \leq 21$ and $26 \leq IEQ \leq 50$), IMINRE contains a minimum firing range (not used for non-firing ground equipments). For air sensors ($51 \leq IEQ \leq 55$), IMINRE contains a minimum detection range. The array is not used for other values of IEQ. The following subroutines use IMINRE:

AIRGRND
CBTVAL
EQINP
FIRALO
FIRSORT
FORMAIN
GENFIR
RADAR
WTSUB

3.85 MODE

3.85.1 English-language Definition

a) Term Name:

Mode

b) Term Description:

Mode of operation has specific meaning for CATTS ground equipment. Air equipment, although using the same arrays, does not have modes of operation, per se.

In CATTS, men and ground equipment are always operating in one of eight "modes." These modes are characterized by a rate of fire, a rate of movement, an ammunition type, and a personnel vulnerability class for each equipment type. Each time step, each equipment type in each unit is fractionally distributed over the eight modes, based on the current tactical situation for that unit. The modes, with exceptions which will be explained later, are user-defined by means of the input tables, so that a proper input definition of the modes will allow virtually any desired combination of movement, firing, and vulnerability to be achieved, and allow different combinations for each equipment in the unit.

An exception is that mode one is assumed to be a suppressed mode and should always be so defined by the input. In fact, the suppression of the fraction of the unit is accomplished by placing that fraction of personnel and equipment into mode one after initially distributing all personnel and equipment over the modes. The mode distribution selected for an equipment may place as many additional pieces of equipment into the suppressed mode one as is desired. For example, in a mechanized infantry unit facing an armored unit, it would be possible, and might be desirable, to define the mode inputs in such a way that anti-armor weapons (and the personnel manning them) became very active, while other equipment (and related personnel) became relatively or completely suppressed.

A set of up to 80 different mode distribution vectors may be specified by input and stored for use in the model. The particular vector of this set that is used with an equipment type of a unit at any given time is determined by three factors: the operational state of the unit, the nature of different enemy units that the equipment will be employed against if it is a weapon, and the proximity of enemy units. The first two factors are embodied in a mode selection code. The third factor involves a mode-selection range. See the definition of the term "break range".

By proper use of the three mode distribution vectors associated with the mode selection code, unit I can be made to change its manner of using equipment as it approaches the enemy without having to change its operational state.

As the various modes of operation of different equipment types are determined, information on the vulnerability of operating personnel in unit I is accumulated, thus providing a vulnerability profile of unit I that can be used in the next step in computing the effects of enemy fire on unit I.

The general definition for the eight modes differs for direct and indirect fire weapons (equipment codes 1 and 2 respectively) and support fire weapons (equipment code = 3). For direct and indirect fire weapons, the modes are defined as follows:

MODE	GENERAL DEFINITION
1	Suppressed, not moving, not firing, not vulnerable
2	moving at top speed, not firing
3	moving fairly fast, not firing
4	moving fairly slow, not firing
5	moving at minimum speed, not firing
6	not moving, firing at sustained rate'
7	not moving, firing at high rate
8	not moving, not firing, highly vulnerable

For support fire weapons, the modes are defined as follows:

MODE	GENERAL DEFINITION
1	suppressed
2	moving, not firing

MODE	GENERAL DEFINITION
3	close support fire, normal
4	close support fire, final protective fires
5	interdictory fire
6	general support for op groups
7	counterbattery fire
8	general support fire

Non-firing equipments are treated as direct fire weapons except that modes 6 and 7 do not have an ammunition type or firing rate assigned.

c) Term References:

There are four attributes describing mode of equipment operation: ammunition type to be used, personnel vulnerability class, rate of fire, and rate of movement. The equipment mode concept is used throughout the model.

3.85.2 Programmatical Definition

a) Term Name:

IAMTE(IEQ,IMODE)
IPVCE(IEQ,IMODE)
ROFE(IEQ,IMODE)
ROME(IEQ,IMODE)

B) Term Description:

Four arrays combine to define each of the eight modes for ground equipments. These arrays are data base input variables that are defined in the equipment input deck. They have entirely different meanings for air equipments, and those definitions are included below.

IAMTE (IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0, ammunition type used by weapon type IEQ in mode of operation IMODE (0implies no ammunition used).
For aircraft (IEQCOD(IAC) = -1),
IMODE = 1 takeoff delay time for this aircraft (min.)
2 landing delay time for this aircraft (min.)
for air ordnance (IEQCOD(IAC) = -2),

IMODE = 1 number of ammunition type for this equipment, if any
2 number of rounds in a standard load of that ammunition type
3 rate of fire of this equipment (rounds/min)
4 number of drops per pass (applies to bombs)
5 distance between drops (meters)
6 dud probability times 100
7 kill probability times 100 for bridge type 1
8 kill probability times 100 for bridge type 2

IPVCE (IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0), personnel vulnerability class associated with each mode IMODE of equipment type IEQ. For air equipment other than an aircraft (IEQCOD (IEQ) = -2 or -3), equipment number of the IMODE-th allowable aircraft type on which IEQ may be used. (Hence, a maximum of eight aircraft per equipment type.) The first zero encountered implies no other allowable aircraft.

ROFE (IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0), rate of fire of weapon type IEQ in each mode IMODE (rounds/minute)
for aircraft (IEQCOD(IEQ) = -1),
IMODE = 1 fuel expenditure for losing altitude (lb/meter)
2 fuel expenditure for gaining altitude (lb/meter)
3 fuel expenditure at minimum speed, minimum load, best pressure density (lb/min)
4 fuel expenditure at cruise speed (lb/min)
5 fuel expenditure at maximum speed (lb/min)
6 ratio of fuel expenditure rate at maximum load to fuel expenditure rate at minimum load
7 ratio of fuel expenditure rate at worst pressure density to fuel expenditure rate at best pressure density

8 not used

For air ordnance (IEQCOD(IEQ) = -2),

IMODE = 1 fraction of personnel in personnel vulnerability class 1 (standing)and within target area who are killed by this equipment

2 fraction of personnel in personnel vulnerability class 2 (crouching) and within target area who are killed by this equipment

3 fraction of personnel in personnel vulnerability class 3 (prone) and within target area who are killed by this equipment

4 fraction of equipment with IEQCLS = 1 and within target area which is damaged by this equipment

5 fraction of equipment with IEQCLS = 2 and within target area which is damaged by this equipment

6 fraction of equipment with IEQCLS = 3 and within target area which is damaged by this equipment

7 fraction of equipment with IEQCLS = 4 and within target area which is damaged by this equipment

8 fraction of equipment with IEQCLS = 5 and within target area which is damaged by this equipment

ROME (IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0), rate of movement of weapon type 1 in MODE IMODE (unobstructed).

For aircraft (IEQCOD(IEQ) = -1,

IMODE = 1 Maximum load aircraft can carry (pounds) at best modeled pressure density (PDBEST)

= 2 maximum altitude of aircraft (meters)

= 3 minimum speed of aircraft (meters/minute)

= 4 cruise speed of aircraft (meters/minute)

= 5 maximum speed of aircraft (meters/minute)

= 6 maximum load aircraft can carry (pounds) at worst modeled pressure density (PDWORST) the correct negative value will insure that an aircraft cannot fly at pressure densities below its capability

= 7 poorest meteorological visibility in which aircraft can continue its mission (meters)

= 8 not used

For equipment other than aircraft (IEQCOD(IEQ) = -2 or -3,

IMODE = 1 weight of equipment (pounds) including standard ammunition load

= 2 minimum aircraft speed (meters/minute) at which equipment can be used

= 3 maximum speed at which equipment can be used (meters/minute)

= 4 minimum altitude at which equipment can be used (meters)

= 5 maximum altitude at which equipment can be

= 6 for sensors, not used; for ordnance, road crater radius (meters) against road type 1

= 7 for sensors, not used; for ordnance, road crater radius (meters) against road type 2

= 8 for sensors, not used; for ordnance, road crater radius (meters) against road type 3

c) Term References:

These arrays are all indexed first on equipment number, IEQ, where ($1 \leq IEQ \leq 80$), second on mode number, IMODE where ($1 \leq IMODE \leq 8$). The normal procedure in using mode information is to loop through all modes of operation in calculating overall movement rate, firing rate, and personel vulnerability class distribution. The mode distribution vector allocates a percentage of personnel and equipment to each of the eight modes. The fraction of the unit suppressed reduces the number of personnel and equipment in each of modes 2 through 8, and adds those fractions to mode 1.

Subroutines FIRALO,ORGFIR,SPTALO, and AMOVUL are the primary users of these arrays for ground equipments. Subroutines ADW,AIRMOV,OTHRDMG and DIDITHIT are among the principal subroutines using the air ordnance data stored in these arrays.

3.86 MODEL TIME VERSUS REAL TIME

3.86.1 English-language Definition

a) Term Name:

Model Time Versus Real Time

b) Term Description:

In CATTS, the passage of time is recorded by a number referred to as "model time". It is set to zero at the beginning of a simulation and subsequently indicates how many units of simulated time have passed since the beginning of the simulation. The term "real time" means the indicated clock time and not the time that the computer has taken to execute the simulation. There is no direct connection between simulated time and the actual time taken to carry out the computations. The controlling factors in determining the computation time are the number of events that occur and the complexity of the computations in the mathematical model.

As a rule, CATTS runs in near real-time, where the ratio of "model time" to "real time" is close to 1:1. The mathematical model is a timestep model, with "model time" timesteps of one minute (except for air units, which have quarter-minute steps). The variable MNSECNDS is the minimum number of "real time" seconds which must pass before a new model timestep is begun. If the number of events and/or the complexity of computations in the mathematical model during a "model time" timestep are small, then "model time" will be faster than a timestep measured in "real time". When this occurs, if MNSECNDS is set equal to 60, the model delays to allow "real time" to catch up to "model time" before the next timestep is entered. However, if MNSECNDS is set equal to 0, the model will execute timesteps as quickly as the number of events and/or the complexity of computations will allow. On the other hand, if the number of events and/or the complexity of computations in the mathematical model during a "model time" timestep are large, then "model time" will, most likely, be slower than a timestep measured in "real time". Typically, the number of events and the complexity of computations in the mathematical model during a "model time" timestep are sufficiently few to allow "model time" to be less than or equal to "real time".

c) Term References:

None

3.86.2 Programmatical Definition

a) Term Name:

ITIME

ITMSTRT

NDAYE

NHOURE

NMINE

TIME

B) Term Description:

ITIME	-	Same as TIME, only an integer number.
ITMSTRT	-	Time, in minutes, from day 0, hour 0, and minute 0. It is calculated from NDAYE, NHOURE, and NMINE.
NDAYE	-	Calendar day of the first day of the game (1-31).
NHOURE	-	Number of elapsed hours at the start of the game since midnight (0-23).
NMINE	-	Number of elapsed minutes in the beginning hour of the game (0-59).
TIME	-	"Model time", in minutes, beginning with the minute input via the data base (a floating point number).

ITIME is a model generated variable that is used in subroutines INPUT and FORMAIN. ITMSTRT is a model generated variable that is used in subroutine INPUT. NDAYE, NHOURE, and NMINE are both data base input variables and model generated variables. As data base input variables, NDAYE, NHOURE, and NMINE are defined in the namelist input deck, and, as model generated variables, they are used in subroutine FORMAIN.

TIME is both a data base input variable and a model generated variable. As a data base input variable, TIME is defined in the first input deck, and, as a model generated variable, it is used in subroutine FORMAIN.

c) Term References:

"Model time" is represented in the simulation by the variable TIME. TIME appears in the following routines:

FORMAIN
INPUT

3.87 MOUNTED/DISMOUNTED

3.87.1 English-language Definition

a) Term Name:

Mounted/Dismounted

b) Term Description:

Mounted/dismounted refers to the manner in which personnel within a unit is moving. The unit is considered mounted when no personnel is traveling by foot (i.e., all personnel are riding in vehicles). The mounted/dismounted mode is determined mainly by the number and types of equipment contained within the unit. Certain equipment types(such as rifles) are identified as dismount-only equipment. That is, they are operable only when the personnel is dismounted, and have an assigned rate of movement consistent with foot movement. Other equipment can operate in either mode, depending upon the number of people available to man the equipment. Each equipment is modeled to have three levels of manning:

1) Minimum crew size when operated in mounted mode
2) Minimum crew size when operated in dismounted mode
3) Maximum personnel carrying capacity

Normally, a unit travels in the mounted mode. This occurs if the troop-carrying capacity of all (vehicular) equipment within the unit can accommodate all personnel. Otherwise, leftover personnel are assigned to a dismount-only equipment. This forces the unit to move at a slower (dismounted) rate. Note that a moving unit, in the mounted mode always assigns personnel to vehicular equipment before assigning personnel to dismount-only equipment. Furthermore, moving units which are unable to man all the vehicles are forced to abandon the unmanned vehicles.

Units traveling in the dismounted mode must man some type of dismount-only equipment. If there is insufficient personnel to accomplish this, the unit remains moving at mounted rates (unless interactively commanded to dismount). If all dismount-only equipment is lost, the

unit is forced to move at mounted rates, provided vehicular equipment still remain: this is true even if the unit is commanded interactively to dismount. Units with no equipment will not move, since unit movement depends on manned equipment.

c) Term References:

During each time step, the determination of mounted/dismounted travel is made depending upon the reassignment of personnel in each unit to the remaining equipment in that unit. This is accomplished automatically by the redistribution logic of the math model. The redistribution, however, can be overridden by interactive command and control. Mounted/dismounted is an option on the maneuver control menu which allows a controller to interactively command a unit to travel in a desired fashion. Note that mounted/dismounted is referenced by unit, hence it is an attribute which characterizes a unit's movement.

3.87.2 Programmatical Definition

a) Term Name:

MOUNTED(4)

b) Term Description:

MOUNTED is a bit packed array giving the mounted/dismounted status of each unit. Its values are model generated in subroutine REDIST. A bit carrying zero means that the unit is mounted, and a bit carrying one indicates that the unit is dismounted.

c) Term References:

The mounted/dismounted status bit is referenced by unit number. The array MOUNTED is four words in length providing 128 bit positions, designated bit 0 through bit 127 (bit 0 is the leftmost bit). To obtain the unit's mounted/dismounted status, reference the information stored in the bit position given by $K(0 \leq K \leq 99)$ where K is determined by subtracting one from the unit number. The array MOUNTED is used in the following subroutines:

FORMAIN
LOWALRT
REDIST

3.88 MOVEMENT CODE (OPERATIONAL GROUPING)

3.88.1 English-language Definition

a) Term Name:

Movement Code (Operational Grouping)

b) Term Description:

Movement code is an integer which distinguishes the sixteen different ways an operational grouping may move. The manner of movement includes, no movement, movement under various conditions of engagement, movement to a specific point or in a specific direction, movement along routes, and movement toward a point relative to a friendly or enemy unit. Associated with each value of the movement code are data values which facilitate the desired manner of movement. Normally, a sequence of movement codes is achieved as various situations arise during the simulation. Specifically, the movement code is changed whenever the operational grouping arrives at its destination or encounters the enemy. Furthermore, the movement code can be manipulated interactively by using the maneuver control menu.

c) Term References:

Movement code is an attribute of operational grouping and is thus referenced by operational grouping number. Movement code is used by all major functions of the CATTS math model.

3.88.2 Programmatical Definition

a) Term Name:

MTCDOG(IOPG)

b) Term Description:

MTCDOG(IOPG) is a data base input variable, an interactively generated as well as a model generated variable. As a data base input variable, it is defined in the operational group input deck; as an interactively generated variable, it is created from the maneuver menu; and, as a model generated variable, it is used by subroutines ARRIVE, CHGOPN,

DEPLOC, FWDLIN, LEAVEOG, NEWENG, NEWFWDUN, NEWMOV, OGLOC, RMVOPOG, TASKORG, and UNI2OG. MTCDOG(IOPG) is an integer array containing a code describing the manner in which operational grouping IOPG ($1 \leq IOPG \leq 20$) is moving. The initial movement code is pre-defined by input, but is changed according to situations (notably unit arrival at destinations and enemy engagements) which arise during the simulation. MTCDOG can also be manipulated by maneuver command and control. Currently the code can be a number between one and sixteen inclusively:

MTCDOG(IOPG)

Movement code of operational grouping IOPG:

1 - Normally engaged

2 - Withdrawing

3 - Deploying (not in position)

4 - Deployed (in position waiting for other units)

5 - Moving in fixed direction

6 - Moving along route

7 - Halted

8 - Moving toward fixed point

9 - Moving toward point relative to friendly operational grouping

10 - Moving toward point relative to enemy operational grouping

11 - Moving toward point relative to friendly engagement FEBA

12 - Moving toward point relative to enemy engagement FEBA

13 - Moving toward point relative to friendly unit

14 - Moving toward point relative to enemy unit

15 - Deploying while not engaged (not in position)

16 - Deployed while not engaged (in position waiting for other units).

c) Term References:

MTCDOG is indexed by operational grouping number IOPG, where IOPG is an integer ranging from one to twenty. It is used by the following subroutines:

ADJDIR	OGINP
ARRIVE	OGLOC
CHGCRT	OGLOC2
CHGOPN	OG2UNI
DEPLOC	OPPLAN
DIRMOV	PREMOV
FWDLIN	RMOPGP
LEAVEOG	STATREP
MOVMNT	TASKPRG
NEWENG	UNI2OG
NEWFWDUN	UN2FEB
NEWMOV	WTHDRW
OGDIR	

3.89 MOVEMENT CODE (UNIT)

3.89.1 English-language Definition

a) Term Name:

Movement Code (Unit)

b) Term Description:

Movement code is an integer which distinguishes the sixteen different ways a unit may move. The manner of movement includes, no movement, movement under various conditions of engagement, movement to a specific point or in a specific direction, movement along routes, and movement toward a point relative to a friendly or enemy unit. Associated with each value of the movement code are data values which facilitate the desired manner of movement. Normally, a sequence of movement codes is acheived as various situations arise during the simulation. Specifically, the movement code is changed whenever the unit arrives at its destination, encounters the enemy, or changes its operational state. Furthermore, the movement code can be manipulated interactively by using the maneuver control menu.

c) Term References:

Movement code is an attribute of unit and is thus referenced by unit number. Movement code is used by all major functions of the CATTS math model.

3.89.2 Programmatical Definition

a) Term Name:

MVTCD(100)

b) Term Description:

MVTCD(IU) is a data base input variable, an interactively generated variable, as well as a model generated variable. As a data base input variable, it is defined in the unit input deck; as an interactively generated variable, it is created from the maneuver menu and, as a model generated variable, it is used in subroutines AIREVENT (for air units only), AIRMOV (for air units only), ANYFOE, ARRIVE,

CHGOPN, DEPLOC, ENCTR, FWDLIN, MANEUVER, NEWENG, OBSDELAY, OBSUPDATA, OG2UNI, OPPLAN, OVERUN, REL2FWDU, STEP, TASKORG, and WTHDRW.

MVTCD(IU) is an integer array containing a code describing the manner in which unit IU ($1 \leq IU \leq 100$) is moving. The initial movement code is pre-defined by input, but is changed according to situations (notably unit arrival at destinations and enemy engagements) which arise during the simulation. MVTCD can also be manipulated by maneuver command and control. Currently the code can be a number between one and sixteen inclusively:

MVTCD (IU)	Movement code of Unit IU: 1 - Normally engaged 2 - Withdrawing 3 - Deploying (not in position) 4 - Deployed (in position waiting for other units) 5 - Moving in fixed direction 6 - Moving along route 7 - Halted 8 - Moving toward fixed point 9 - Moving toward point relative to friendly operational grouping 10 - Moving toward point relative to enemy operational grouping 11 - Moving toward point relative to friendly engagement FEBA 12 - Moving toward point relative to enemy engagement FEBA 13 - Moving toward point relative to friendly unit 14 - Moving toward point relative to enemy unit 15 - Deploying while not engaged (not in position) 16 - Deployed while not engaged (in position waiting for other units).

c) Term References:

MVTCD is indexed by unit number IU, where IU is an integer ranging from one to one hundred. It is used by the following subroutines:

ADJDIR	FIRELG	OBSDELAY	STATREP
ADJROM	FRNTGS	OBSUPDAT	STEP
ANYFOE	FWDLIN	OGDIR	STLKUP
AREA	HUMAN	OGLOC	TASKORG
ARRIVE	INPUT	OG2UNI	UNBLOK
CHGCRT	LATDST	OPPLAN	UNINP
CHGOPN	MANEUVER	OVERUN	UNI2OG
DEPLOC	MOVE	REL2FWDU	UN2FEB
DIRMOV	MOVMNT	ROADCHK	USEFUEL
ENCTR	NEWENG	SAVE	WTHDRW
FINDFO	NEWFWDUN	SAVEINP	
FIRALO	NEWMOV	SAVEOLD	

NOTE: MVTCD has an entirely different definition unrelated to movement when used by the following subroutines in the air module logic:

AIREVENT
AIRMOV

3.90 MOVEMENT RATE

3.90.1 English-language Definition

a) Term Name:

Movement Rate

b) Term Description:

Movement rate is defined to be the speed at which a unit is traveling. It is determined each time step by examining the movement rates of every equipment type in the unit, and choosing the rate of the slowest moving equipment type. The determination of movement rate involves consideration of the following for each type of equipment in the unit:

1) the distribution of modes of operation of the equipment type
2) input-defined desired speeds at which the equipment type should operate in its various modes
3) the amount of suppression experienced by the unit
4) environmental constraints

Further information on movement rate is available in this document; see definitions of the following terms:

1) input equipment movement rate
2) maximum equipment movement rate
3) computed equipment movement rate

c) Term References:

Movement is an attribute of unit, thus it is referenced by unit number (an integer between 1 and 100 inclusively). Movement rate is computed by the ground fire module for use by the movement and target acquisition modules. The movement logic takes the unit's rate of movement, direction of movement, and present location to calculate where the unit will be located at the next time step. Target acquisition is concerned with moving units because while units have a positive rate of movement, their observation capabilities (i.e., ability to detect the enemy) are degraded. Also, if a unit is moving, its chances of being detected is enhanced. The fuel consumption module also references movement rate to determine fuel expenditures by the unit during the timestep.

3.90.2 Programmatical Definition

a) Term Name:

ROMU(100)

b) Term Description:

The floating point array contains the movement rate in meters per minute for every active unit. Computation of movement rate for each unit involves looping through each equipment type in the unit, computing the movement speed of the equipment type, and determining whether this speed is slower than any previously computed speed. If it is slower, the unit movement rate is set to this speed, otherwise the loop continues examining the next equipment type, until the loop has considered all equipment type. Upon loop completion, the unit's movement rate will have been set to the rate of the slowest moving equipment type.

ROMU(IU) = rate of movement of unit IU ($1 \leq IU \leq 100$) in meters per minute

ROMU(IU) is a model generated variable that is used in subroutines AIREVENT (for air units only), AIRMOV (for air units only), ARRIVE, CRDLIC (no menu was ever written which would cause this subroutine to be called), DIRMOV, FINDWA (this subroutine is no longer called since it does not work properly subsequent to the addition of detection logic to subroutine ENCTR), FIRALO, MOVE, MOVMNT, OPPLAN, OVERUN, SPTALO, STEP, and WPVEL.

c) Term References:

The array ROMU(IU) is indexed by unit number IU, where IU is an integer ranging from 1 to 100 inclusively. ROMU is used in the following subroutines:

ARRIVE	LOWALRT
CHGCRT	MOVE
CRDLIC	MOVMNT
DIRMOV	OPPLAN
ENCTR	OVERUN
FINDWA	SPTALO
FIRALO	STEP
FORMAIN	WPVEL

3.91 NAVAL OPERATION

3.91.1 English-language Definition

a) Term Name:

Naval Operation

b) Term Description:

Naval operation is represented by a special unit called battleship which has the following attributes:

1) the unit must be located in water
2) the unit will remain stationary throughout the simulation (interactive maneuver control is not available for this unit)
3) the unit will have only 203 MM Howitzers; these weapons can be fired interactively via the fire control menu
4) the unit will contain enough personnel to man the weapons.

c) Term References:

Naval operation is referenced basically by the firing logic, since this special unit has no other capabilities other than to fire its weapons against the enemy. Note that it is also vulnerable to fire and may sustain casualty and damage.

3.91.2 Programmatical Definition

(Not applicable)

3.92 NEXT HIGHER COMMAND

3.92.1 English-language Definition

a) Term Name:

Next Higher Command

b) Term Description:

A unit may or may not be subordinate to another unit. Those which are subordinate have a next higher command. The next higher command is a "unit" to which the unit must report to directly. Note that the next higher command "unit" may in fact not be a unit at all. The following unit numbering convention is established:

1. Units 1-100 are legitimate units
2. Units 101-120 are in reality operational groupings 1-20
3. Units 121-135 are fake (i.e., nonexistent) red units
4. Units 136-150 are fake (i.e., nonexistent) blue units

Unit number 101-150 provide a representation for higher commands which are implicitly assumed to exist though not explicitly represented.

The concept of next higher command is used by the CATTS math model for two basic reasons. First of all, control measures associated with a given "unit" can affect each of its subordinate units. This control measure alerts must be generated not only for a given unit, but for all of its subordinate units. Control measures associated with operational groupings (i.e., unit numbers 101-120) will cause alert messages to be generated only for units belonging to the corresponding operational grouping. The second reason for establishing a hierarchy of commands relates to the interactive capability of resupply. Unit's may be resupplied with equipment and/or personnel from higher "non-existent" command units. These fake units are not represented at all in the math model (i.e., they have no attributes like equipment, personnel, etc., defined); they serve as a (possibly infinite) source of supplies to be allocated at the discretion of the instructor.

c) Term References:

Next higher command is an attribute of units, hence it is referenced by specifying the unit number. Next higher command is used in conjunction with control measure interactions. All control measures are associated with units, and any interaction by these units that trigger control measure alerts require that alerts be generated also for subordinate units. By establishing the hierarchy of unit command (via input), subordinate units can be readily identified. The task organization logic must update the next higher command specification whenever it creates a new operational grouping or adds or deletes units from existing operational groupings. Units which become defunct require a similar update. The interactive capability of resupply indirectly references the next higher command concept. A fake unit can be established as a higher command concept. A fake unit can be established as a higher command unit from which new materials and/or personnel can be obtained for depleted units.

3.92.2 Programmatical Definition

a) Term Name:

NXHGCM(150)

b) Term Description:

The integer array NXHGCM is both a data base input variable and an interactively generated variable. As a data base input variable, it is defined in the listing and next higher command input deck, and, as an interactively generated variable, it is created from the tasking organization menu. NXHGCM gives, for each unit, the unit number of its next higher command. The first 100 elements of the array contain data for legitimate units. The next 20 elements contain information for command units representing the model's operational groupings. The remaining 30 elements contain data for fake red and blue command units, units which exist implicitly, but have no explicit attributes defined for them. The data contained in the Ith array element is an integer ranging from 0 to 150 inclusively indicating for the Ith unit, which unit is the next higher command unit. If zero is specified, the Ith unit has no next higher command unit.

NXHGCM(IU) The unit which is next higher command for Unit IU, $1 \leq IU \leq 150$.

c) Term References:

The integer array NXHGCM is indexed by unit number *IU*, where *IU* ranges from 1 to 150 inclusively. Note that the indices 1 through 100 are numbers of legitimate units, while indices 101 through 150 are artificial unit numbers. The array NXHGCM is used by the following subroutines:

ADD2LIST	INPUT
CK4XING	JOINOG
CRDLIC	LEAVEOG
DTECTPRB	MKUNLIST
FIXLIST	TASKORG

3.93 OBSERVATION POST

3.93.1 English-language Definition

a) Term Name:

Observation Post

b) Term Description:

An observation post is designated in the model by setting the unit arm/branch/duty value for a specific unit to be negative. This causes the observation post display symbol to appear for that unit at the unit's location when the observation post button is selected. The symbol for observation post is .

c) Term References:

The foreground graphics display programs refer to the unit arm/branch/duty of each unit, unit number being a direct reference.

Those units having a negative value have the observation post symbol displayed at the unit's location. Op groups and adjacent units do not have observation posts assigned to them.

3.93.2 Programmatical Definition

a) Term Name:

ITPPL(IU)

b) Term Description:

ITPPL(IU) is a data base input variable that is defined in the unit input deck. An observation post is designated for unit IU in the model by setting array ITPPL(IU) to a negative value as part of the data base inputs. This value is never altered for units. The negative value designation for an observation post is not used for op groups or adjacent units ($101 \leq IU \leq 150$).

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ITPPL array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ITPPL:

FORRAM

UNINP

3.94 OBSTACLES

3.94.1 English-language Definition

a) Term Name:

Obstacles

b) Term Description:

Obstacles in the CATTS math model are defined by a series of line segments having the following attributes:

1) the army to which the obstacle belongs
2) obstacle type
3) width
4) the X-Y coordinates of the endpoints of the line segments
5) obstacle name
6) number of line segments comprising the obstacle

Currently obstacles must be pre-defined by input. A maximum of fifty obstacles may exist simultaneously. Obstacles serve to restrict the movement of units and operational groupings, and may inflict casualty and damage as well.

c) Term References:

Obstacle is an entity referenced by an identifying integer between one and fifty inclusively. Obstacles are used by the movement logic to delay units, and to inflict damage and/or casualty.

3.94.2 Programmatical Definition

a) Term Name:

IOBRB(50),IOBSTYPE(50),IOBWIDTH(50),
IOBX(6,50),IOBY(5,50),MOFNAME(3,50),NSEGMT(50)

b) Term Description:

The following arrays are data base input variables that are devined in the mine obstacle fortification deck. They describe the set of obstacles in the CATTS math model:

IOBRB (IOBS)

Code designating which army the IOBSth (1 < = IOBS < = 50) obstacle belongs to:

= 0 either army

= 1 red army

= 2 blue army

IOBSTYPE(IOBS)

The type of the IOBSth (1 < = IOBS < = 50) obstacle where:

= 1 crater field

= 2 general mass obstacle

= 3 minefield

= 4 lake

= 5 waterway (canel, river, etc.)

= 6 concertina barrier

= 7 fixed wall (barrier)

= 8 ditch

= 9 ravine

=10 cliff

IOBWIDTH(IOBS)

The width (when applicable) across the IOBSth (1 < = IOBS < = 50) obstacle

IOBS (I,IOBS)

X coord. of endpoint of the Ith (1 < = I < = 6) line segment describing the IOBSth (1 < = IOBS < = 50) obstacle

b) Term Description:

The following arrays are data base input variables that are devined in the mine obstacle fortification deck. They describe the set of obstacles in the CATTS math model:

IOBRB (IOBS)

Code designating which army the IOBSth (1 < = IOBS < = 50) obstacle belongs to:

= 0 either army

= 1 red army

= 2 blue army

IOBSTYPE(IOBS)

The type of the IOBSth (1 < = IOBS < = 50) obstacle where:

= 1 crater field

= 2 general mass obstacle

= 3 minefield

= 4 lake

= 5 waterway (canel, river, etc.)

= 6 concertina barrier

= 7 fixed wall (barrier)

= 8 ditch

= 9 ravine

=10 cliff

IOBWIDTH(IOBS)

The width (when applicable) across the IOBSth (1 < = IOBS < = 50) obstacle

IOBS (I,IOBS)

X coord. of endpoint of the Ith (1 < = I < = 6) line segment describing the IOBSth (1 < = IOBS < = 50) obstacle

IOBY (I,IOBS)

Y coord. of endpoint of the Ith (1 < = I < = 6) line segment describing the IOBSth (1 < = IOBS < = 50) obstacle

MOFNAME (I,IOBS)

Alphanumeric name of the IOBSth (1 < = IOBS < = 50) obstacle each name is stored in at most three full words (1 < = I < = 3)

NSEGMT (IOBS)

The number of line segments comprising the IOBSth (1 < = IOBS < = 50) obstacle

c) Term References:

Each obstacle is referenced by its identifying integer, which ranges from one to fifty inclusively. If there are less than fifty obstacles in the model, the total number of obstacles is given by NOB ($1 \leq NOB \leq 50$); the identifying integers would then range from one to NOB inclusively. The above arrays as well as the variable NOB must be pre-defined via input. Values in these arrays remain constant throughout the simulation (cannot be changed). The obstacle arrays and variables are used in the following subroutines:

BRCHPATH	NEAROBS
CMAIN	OBFMINP
ENGUPDAT	OBSCHECK
FORRAM	OBSDELAY
INPUT	OBSTACLE
LINESEG	OBSUPDAT
LOWALRT	OBSWIDTH
MINEFLDS	

3.95 OBSTACLE DELAY TIME

3.95.1 English-language Definition

a) Term Name:

Obstacle Delay Time

b) Term Description

Obstacle delay time is associated with the type of obstacle encountered by a unit. A unit is detained a minimum of two minutes whenever it runs into an obstacle. Further delay time is added depending on the following factors:

1) the type of obstacle encountered
2) the distance to be traversed in order to cross over the obstacle
3) the availability of engineering support
4) the number of personnel remaining in the unit to help reduce the obstacle.

c) Term References:

The assessment of delay time is referenced by the type of obstacle encountered. This delay time is used by the movement function to simulate obstacle interaction. It models the time required to reduce the obstacle by unit personnel and engineering support.

3.95.2 Programmatical Definition

a) Term Name:

TASK(10)

b) Term Description:

TASK is a data base input variable that is definedin the namelist input deck. It is an array containing the delay information for each of the ten different types of obstacles that a unit may encounter. For an obstacle of type IOBST we have:

TASK (IOBST)

Time delay assessed against a unit by an obstacle of type IOBST (1 < = IOBST < = 10) expressed in terms of manhour per meter

The array TASK must be pre-defined via NAMELIST input. Its values remain constant throughout the simulation.

c) Term References:

TASK is indexed by obstacle type IOBST, where IOBST is an integer ranging from one to ten inclusively. TASK is used by the following subroutine:

ENGRSPT

3.96 OBSTACLE STATUS

3.96.1 English-language Definition

a) Term Name:

Obstacle Status

b) Term Description:

Obstacle status is an attribute of the entity unit. It describes the unit's movement status with respect to obstacles. The obstacle status of a unit governs completely the sequence of events requried to cross over an obstacle. This entails the initial encounter, the positioning of the unit in front of the obstacle to endure the delay, and the travel across the obstacle to a point that cleans the obstacle entirely.

c) Term References:

Since obstacle status is an attribute of unit, it must be referenced by unit number. Obstacle status is used mainly by the movement logic to control the movement of the unit when it must interact with an obstacle. This status determines when the unit must remain delayed, when it can start it moves across the obstacle, and finally when the unit has cleared the obstacle entirely.

3.96.2 Programmatical Definition

a) Term Name:

IOBSTATU(100)

b) IOBSTATU is a model generated variable that is used by subroutines ENGRSPT, ENGUPDAT, FORMAIN (data statement to initialize), MANEUVER, OBSDELAY, and OBSUPDAT. IOBSTATU is an integer array containing a code for each unit. The code ranges from zero to six and is defined as follows:

IOBSTATU(JU) = Status of JUth ($1 \leq JU \leq 100$) unit with respect to obstacles:

= 0 Unit is not stopped by an obstacle

= 1 Unit is stopped at an obstacle, and engineering support is available

= 2 Unit is traversing through an obstacle

= 3 Unit is stopped at an obstacle waiting for engineering support

= 4 Unit is stopped by an obstacle requiring the construction of a bridge without the aid of engineering support

= 5 Unit is stopped by an obstacle requiring the construction of a bridge with the aid of engineering support

= 6 Unit is stopped temporarily in order to prepare for a bridge crossing operation

c) Term References:

IOBSTATU is indexed by unit number. It is used in the following subroutines:

DIRMOV
ENCTR
ENGRSPT
ENGUPDAT
FORMAIN
MANEUVER
MOVE
OBSDELAY
OBSUPDAT
OPPLAN

3.97 OBSTACLE TYPE

3.97.1 English-language Definition

a) Term Name:

Obstacle Type

b) Term Description:

Obstacle type is an attribute of obstacles which distinguishes the various kinds of obstacles modeled by the CATTS software. The distinction lies mainly in the amount of delay time assessed against a unit when it encounters an obstacle. Currently ten types of obstacles have been established:

1) crater field
2) general mass obstacle
3) minefield
4) lake
5) waterway (canal, river, etc.)
6) concertina barrier
7) fixed wall barrier
8) ditch
9) ravine
10) cliff

Obstacle types one through six are termed area obstacles. That is, they are mathematically represented either as:

1) simple convex polygons, or
2) the union of a set of rectangles generated from a given width and a series of connecting line segments which do not close to from a polygon nor intersect each other except at the connecting endpoints.

Obstacle types seven through ten are called linear obstacles because they are simply comprised of a series of connecting straight line segments.

c) Term References)

Every obstacle has an identifying integer (a number between one and fifty inclusively). To reference the type of a given obstacle, the identifying obstacle integer must be known. For each obstacle number, there is a corresponding number (between one and ten inclusively) relating the obstacle type. Obstacle type is used by the movement function to assess delay times against units, reduce delay times if engineering support is available, and to plan routes in order to travel across obstacles.

3.97.2 Programmatical Definition

a) Term Name:

IOBSTYPE(50)

b) Term Description:

IOBSTYPE is a data base input variable that is defined in the mine obstacle fortification input deck. It is an integer array containin a code for each obstacle defined in the model. The code is given as follows:

IOBSTYPE(IOBS)

The type of the IOBSth (1 < = IOBS < = 50) obstacle where:

= 1 crater field
= 2 general mass obstacle
= 3 minefield
= 4 lake
= 5 waterway (canal, river, etc.)
= 6 concertina barrier
= 7 fixed wall (barrier)
= 8 ditch
= 9 ravine
=10 cliff

The array IOBSTYPE is pre-defined by input and remains constant throughout the simulation.

c) Term References:

The array IOBSTYPE is indexed by obstacle number. It is used by the following subroutines:

BRCHPATH	OBFMINP
ENGUPDAT	OBSDELAY
FORRAM	OBSTACLE
LOWALRT	OBSUPDAT
MINEFLDS	OBSWIDTH

3.98 ONROAD/OFFROAD

3.98.1 English-language Definition

a) Term Name:

Onroad/Offroad

b) Term Description

The term onroad refers specifically to whether a given unit is moving along a road. A unit is traveling only if the following two conditions are met:

1) the movement code of the unit must be set to SIX
2) the unit must be located no further than 250 meters away from the center line of the road.

A unit is traveling offroad if either of the above two conditions are not satisfied. Roads are generally free of constraining terrain factors; this allows units to move at a faster rate. The 250 meter limit is an input defined quantity that can be changed readily.

c) Term References:

Onroad/offroad is an attribute of units describing its movement; thus it is referenced by unit number. The onroad/offroad movement status is used by the movement function to develop environmental degradation factors which model the retarding effect of terrain on equipment movement. Offroad units must contend with more terrain factors, hence the movement degradation consumption logic to compute the amount of fuel expended.

3.98.2 Programmatical Definition

a) Term Name:

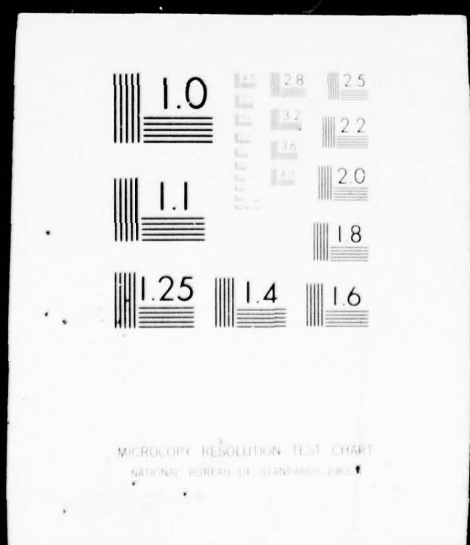
IONROAD(100)

AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156
UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

5 OF 8
AD
A038796

5 OF 8

AD
A038796



b) Term Description

IONROAD is a model generated variable that is used in the main program CMAIN (initialized) and in subroutine ROADCHK. It is an integer array which indicates for each unit whether the unit is traveling on (=1) or off(=0) road. IONROAD(IU)=1 if unit IU satisfies the following conditions:

1) MUTCD(IU)=6 (movement code indicates route movement)
2) the unit location given by {IXY(IU,1).IXY(IU,2)} is within 250 meters of the center line of a road segment

IONROAD(IU)=0 if unit IU fails either of the above conditions.

c) Term References:

IONROAD is indexed by unit number. It is used by the following subroutines:

ADJROM
CMAIN
FORRAM
MOVE
ROADCHK
STATREP
STATREP1
USEFUEL

3.99 OPERATIONAL GROUPING

3.99.1 English-language Definition

a) Term Name:

Operational Grouping

b) Term Description:

Operational grouping is a set of units, functioning collectively to achieve a common objective. Operational groupings can be created in two ways: by user defined input, or by using the interactive command and control task organization menu. Presently a maximum of twenty operational groupings may exist concurrently in the model.

Operational grouping provides the capability of maneuvering a group of units as a single entity. It allows the units to position themselves in formation relative to one another, and move or deploy in these formations to engage the enemy. Furthermore, treating a collection of units as an operational group relieves the controller at the instructor station from the time consuming task of having to issue many interactive commands to several individual units to accomplish a single objective. Note that units are allowed to join or leave operational groups according to situations which arise during the simulation.

c) Term References:

Operational groupings are entities with attributes described by the arrays above. When referencing the attributes of a given operational grouping, its identifying integer must be specified. Operational groupings are referenced and used mainly by the movement, engagement, and command and, control (i.e., task organization, maneuver) modules.

3.99.2 Programmatical Definition

a) Term Name:

DPMVT(20,2)	IDWENG(20)	IREENG(20)	MTCDOG(20)
FECFAC(20)	IFMUN(20)	ITRAVG(20)	MVDTA1(20)
HLFRN(20)	IOGSTAT(20)	IXF(20)	MVDTA2(20)
IDPCOD(20)	IOGTYP(20)	IXYOG(20)	MVDTA3(20)

b) Term Description:

All operational groupings are uniquely defined by the data stored in the arrays listed above. Note that each array is dimensioned by twenty, indicating that the model willallow a maximum of twenty operational groupings to exist simultaneously. The arrays are defined below, with the following convention: the subscripts IOPG and JOPG represent operational grouping number which is an integer between 1 and 20 inclusively; IOPG reference arrays which are input, JOPG reference arrays which have computed values stored in them.

DPMVT (IOPG,J) — Direction faced by operational grouping -stored as sin (J=1) and cos (J=2) - cartesian coordinate system

FECFAC (IOPG) — Frontage expansion - contraction factor for the IOPGth operational grouping.

HLFRN (IOPG) — Normal half-frontage of the IOPGth operational group

IDPCOD (IOPG) — Deployment code of the IOPGth operational grouping
= 0-not deployed and cannot move
1-not deployed and can move
2-technically deployed
3-actually deployed.

IDWENT (IOPG) — Depth of the IOPGth operational grouping when it is engaged.

IFMUN (JOPG)	Forward-most unit of the JOPGth operational grouping in its direction of movement.
IOGSTAT (JOPG)	Status of operational group JOPG IOGSTAT=0 normal op group IOGSTAT=-1 defunct op group
IOGTYP (IOPG)	Characteristic unit type for the IOPGth operational group.
IREENG (IOPG)	If the IOPGth operational grouping is engaged, IREENG is the minimum distance of its forward-most unit from enemy FEBA that would permit automatic transfer of grouping from its current engagement to a closer, new engagement (and then only if forward-most unit's status code is 0). Unless there is a reason for keeping attention of an operational grouping focused on its present engagement, IREENG might often be set to appropriate NGARNG.
ITRAVG (IOPG)	Travel code for the IOPGth operational grouping
IXF (JOPG)	Coordinate of the JOPGth operational grouping in its direction of movement in a system rotated about the fixed origin.
IXYOG (JOPG,J)	X (J=1) and Y(J=2) coordinates of JOPGth operational grouping
MTCDOG (IOPG)	Movement code of the IOPGth operational grouping; see movement codes for unit (MUTCD)
MVDTA1 (IOPG)	Movement data set 1 of the IOPGth operational grouping; see movement data for unit (MVDT1)
MVDTA2 (IOPG)	Movement data set 2 of the IOPGth operational grouping; see movement data for unit (MVDT2)
MVDTA3 (IOPG)	Movement data set 3 of the IOPGth operational grouping; see movement data for unit (MVDT3)

IFMUN, IOGSTAT, and IXF are model generated variables that are used by the Command and Control Module (Task Organization Events Processor).

IXYOG is a model generated variable that is used by the Ground Movement and Engagements Modules.

DPMVT is a data base input variable, an interactively generated variable, as well as a model generated variable. As a data base input variable, it is defined in the operational grouping input deck; as an interactively generated variable, it is created from the maneuver, tasking organization, and unit location menus; and, as a model generated variable, it is used by the Ground Movement and Engagements Modules.

FECFAC, HLFRN, IDPCOD, IDWENG, IOGTYP, and IREENG are both data base input variables and model generated variables. As data base input variables, they are defined in the operational grouping input deck, and, as model generated variables, they are used by the Command and Control Module (Task Organization Events Processor).

ITRAVG, MTCDOG, MVDTA1, MVDTA2, and MVDTA3 are data base input variables, interactively generated variables, as well as model generated variables. As a data base input variable, it is defined in the operational grouping input deck; as an interactively generated variable, it is created from the maneuver menu, and as a model generated variable, it is used in the Command and Control Module (Task Organization Events Processor). MTCDOG, MVDTA1, MVDTA2, and MVDTA3, as model generated variables, are also used by the Ground Movement and Engagements Modules.

c) Term References:

The arrays describing the entity operational grouping are indexed by IOPG (or JOPG), the operational grouping number, which is an integer ranging from 1 to 20 inclusively. These arrays appear in one or more of the following subroutines:

ADJDIR	DIRMOV	NEWENG	OGLOC	SETOGCDS
ANYFOE	FIXOGCDS	NEWFWDUN	OGLOC2	STATREP
ARRIVE	FORRAM	NEWMOV	OGTYPE	STATREP1
CHGCRT	FWDLIN	NGARGN	OG2UNI	TASKORG
CHGOPN	LEAVOG	OGDIR	OPPLAN	UNI2OG
CMAIN	MANEUVER	OGFRNT	PREMOV	UN2FEB
DEPLOC	MOVE	OGHFRONT	REL2FWDU	WPNEFF
DEPLOY	MOVMNT	OGINP	RMVOPGP	WTHDRW

3.100 OPERATIONAL GROUPING TYPE

3.100.1 English-language Definition

a) Term Name:

Operational Grouping Type

b) Term Description:

Every operational grouping has a characteristic type. This type is determined by the most common type of unit contained in the operational grouping. Presently, as many as twenty different types of unit may be defined, hence twenty different types of operational grouping may be specified. The type of an operational group may be specified by user defined input, or automatically determined when the task organization menu is used to create a new operational grouping.

c) Term References:

Type of operational grouping is an attribute of the entity operational grouping, hence type is referenced by specifying the operational grouping number. This number is an integer which ranges from 1 to 20 inclusively. Type is used mainly by the movement and engagement modules. Whenever a new operational grouping has been created interactively by the task organization capability, type is automatically determined by the appropriate software.

3.100.2 Programmatical Definition

a) Term Name:

IOGTYP(20)

b) Term Description:

IOGTYP is both a data base input variable and a model generated variable. As a data base input variable, it is defined in the operational grouping input deck, and, as a model generated variable, it is used by subroutine OGTYPE. Every operational grouping defined in the model has associated with it, a code stored in the integer array IOGTYP, indicating what type of operational grouping it is. This code is an integer between 1 and 20 inclusively, and corresponds to the same code indicating unit type (see the term definition unit type). In fact, the operational grouping type is determined by the type of the most common unit contained in the group - IOGTYP(IOPG)

= characteristic unit type for
the IOPGth ($1 \leq$ IOPG ≤ 20)
operational grouping

c) Term References:

The operational grouping type array IOGTYP is indexed by IOPG, where IOPG is an integer between 1 and 20 inclusively, specifying the operational grouping number. The array IOGTYP is initialized to default values at the beginning of the simulation, and remains constant unless a new operational grouping is created; then the code designating type is added to the appropriate storage location in the array. IOGTYP is used in the following subroutines:

ANYFOE
MOVE
NGARGN
OGDIR
OGFRNT
OGINP
OGTYPE

3.101 OPERATIONAL STATE

3.101.1 English-language Definition

a) Term Name:

Operational State

b) Term Description:

Each unit may assume any one of up to 99 operational states defined by the user (0 through 98). An operational state should represent a method of operation, the general intentions, the condition, or any other convenient characteristic of a unit. Operational states are initially defined via data base input. The operational state of a unit can be changed during the running of an exercise either automically or through command and control. Unit operational states can change automatically in a variety of ways, as summarized in Table 3-2. See the following section for a complete discussion of the program tables referenced above.

Unit operational state is a critical factor in a unit's operation. The operational state is used in determining:

1) the mode distribution vector for all ground equipments other than support fire weapons by selecting the appropriate set of break ranges (see term discussion for "break range"); with the present data base operational state is the sole factor in this determination

2) support fire weapon operation by finding a match in the *Support Fire Table, in terms of:*
   - mode distribution vector
   - support fire "bands" to fire general support (see definition of "bands")
   - support fire "sectors" to fire general support (see definition of "sectors")
   - other automatic support fire allocation indicators

3) the suppression criterion and curve to determine a unit's suppression (see discussion of "Suppression tables in next section)

Table 3-2. Unit Operational State Cha

Unit(s)	Scenario Control		Arrival a
	User Direction	Internal Process	User Direction
Unattached or "lead" unit of an operational grouping.	Change of State Table entry	The appropriate entry in table is located and checked to see if the Change of State Criterion is satisfied. Up to two changes of state in one time interval are allowed.	Movement Code Change Table Operational State Updating Table
Following units of an operational grouping	Operational State Updating Table Change of State Table	If the unit follows its operational group or has had its movement code automatically set to "withdrawing from an engagement" (= 2), this table is searched for an appropriate entry to effect a change in operational state A following unit may be treated just like a "lead" or unattached unit if the user does not care to use the Operational State Updating Table. However automatic changes in movement code must be anticipated.	Operational State Updating Table

ntrol	Arrival at Destination		Engagement by Encounter	
ernal Process	User Direction	Internal Process	User Direction	Internal Process
ppropriate entry ble is located hecked to see if hange of State rion is satis- Up to two es of state in ime interval llowed.	Movement Code Change Table Operational State Updating Table	As a result of movement code change, it may be desirable to change the state of the Unit. If any unit becomes engaged or if a unit is the last unit of an an operational grouping to become deployed, this table is searched for an appropriate entry to cause a change in operational state.	Operational State Updating Table	This table is searched for an appropriate entry to effect a change in operational state.
e unit follows perational group s had its move- code automatica- et to "withdraw- rom an engage- (= 2), this is searched for propriate entry fect a change in tional state lowing unit may eated just like a l" or unattached if the user does are to use the tional State Up- g Table. However atic changes in ent code must ticipated.	Operational State Updating Table	This table is searched for an appropriate entry to effect a change in operational state.	Operational State Updating Table	This table is searched for an appropriate entry to effect a change in operational state.

4) the effect of enemy fire, considering the unit as a target unit (this option is presently not used -- see discussion of Weapons Effects Table) -

Finally, a unit's operational state can be changed interactively by means of the "maneuver control" menu. A list of eleven of the more common operational states, ten for maneuvering units using direct or indirect fire weapons, and one for support fire units. This list is identified in the programmatic description, along with the complete list of currently defined operational states

c) Term References:

3.101.2 Programmatical Definition

a) Term Name:

IOPSTU(IU),FREXOPST(IOPST),NAMEOPST(K,IOPST)

b) Term Description:

The array IOPSTU(IU) is defined simply as

IOPSTU (IU)	Operational state of unit IU. If IU is an air unit, IOPSTU = 1 if IU is on a recommaissance mission and = 2 if IU is on a strike mission.

The variables listed below are directly indexed by operational state number. The tables discussed in the English description also use operational state number along with other factors to determine which data in that table or another table is applicable to the unit.

FREXOPST(IOPST)	fraction of target unit which would normally be exposed to view if it were operating in operational state IOPST
NAMEOPST(K,IOPST)	16 character names associated with op state IOPST K=1 first four characters K=2 second four characters K=3 third four characters K=4 fourth four characters

IOPSTU is a data base input variable, an interactively generated variable, as well as a model generated variable. As a data base input variable, it is defined in the unit input deck; as an interactively generated variable, it is created from the air menu (air units only) and maneuver menu; and, as a model generated variable, it is used in the Ground Movement (Obstacle) and Command and Control (Task Organization Events Processor) Modules, as well as the MAFIA Command and Control Module using data input in the change of state and movement decks.

FREXOPST and NAMEOPST are data base input variables that are defined in the namelist input deck and operational state name input decks, respectively.

The operational states defined presently in the model are listed below. Those asterisked appear on the maneuver control menu.

11 moving/contact*
12 recon in force*
13 attack*
14 displacing*
15 attack stalled
16 exploitation*
17 recon
21 defense/delibrt*
22 defense/hasty*
31 delay*
32 withdraw(vol.)*
33 withdraw(invol.)
34 retire
41 non-firing mort
42 firing-slow mort
43 firing-sus mort
44 firing-max mort
51 fire at band 1
52 fire at band 2
53 fire at band 3

54 fire at band 4
55 fire at N band 1
56 fire at S band 1
57 fire at N band 2
58 fire at S band 2
59 fire at N band 3
61 fire at S band 3
62 fire at N band 4
63 fire at S band 4
64 close suppt-norm
65 close suppt-FPF
66 interdictory
67 counterbattery
68 arty displacing
69 arty non-firing
71 ada-hold
72 ada-tight
73 ada-free
74 ada-displacing
81 combat command
82 reserve command
85 halted/reserve*
86 displacing/resrv
88 adjacent unit

c) Term References:

Array IOPSTU(IU) is a unit attribute, where $1 \leq IU \leq 100$. Arrays FREXOPST(IOPST) and NAMEOPST(K,IOPST) are indexed on operational state number, IOPST, where ($1 \leq IOPST \leq 100$). FREXOPST(100) and NAMEOPST(K,100) are not used, however. NAMEOPST allows 12 characters, 4 per word, so dummy index K ranges from ($1 \leq K \leq 3$). These arrays are used throughout the model.

3.102 OVERLAY

3.102.1 English-language Definition

a) Term Name:

Overlay

b) Term Description:

An overlay consists of a group of programs co-existing in main memory. Since the math model is so large, there is not enough main memory to accommodate the entire model at one time. It therefore, is logically divided into overlays, or segments, that are stored on disc and time-sequenced into main memory for execution. The main math model driver program, FORMAIN, performs this time-sequencing. It is essential in defining each overlay that it be self-contained; that is, no program included in the overlay references any code or data that is not in main memory with that overlay. Every time step, the overlays depicted in Figure 3-4 are time-sequenced.

c) Term References:

None

3.102.2 Programmatical Definition

(Not applicable)

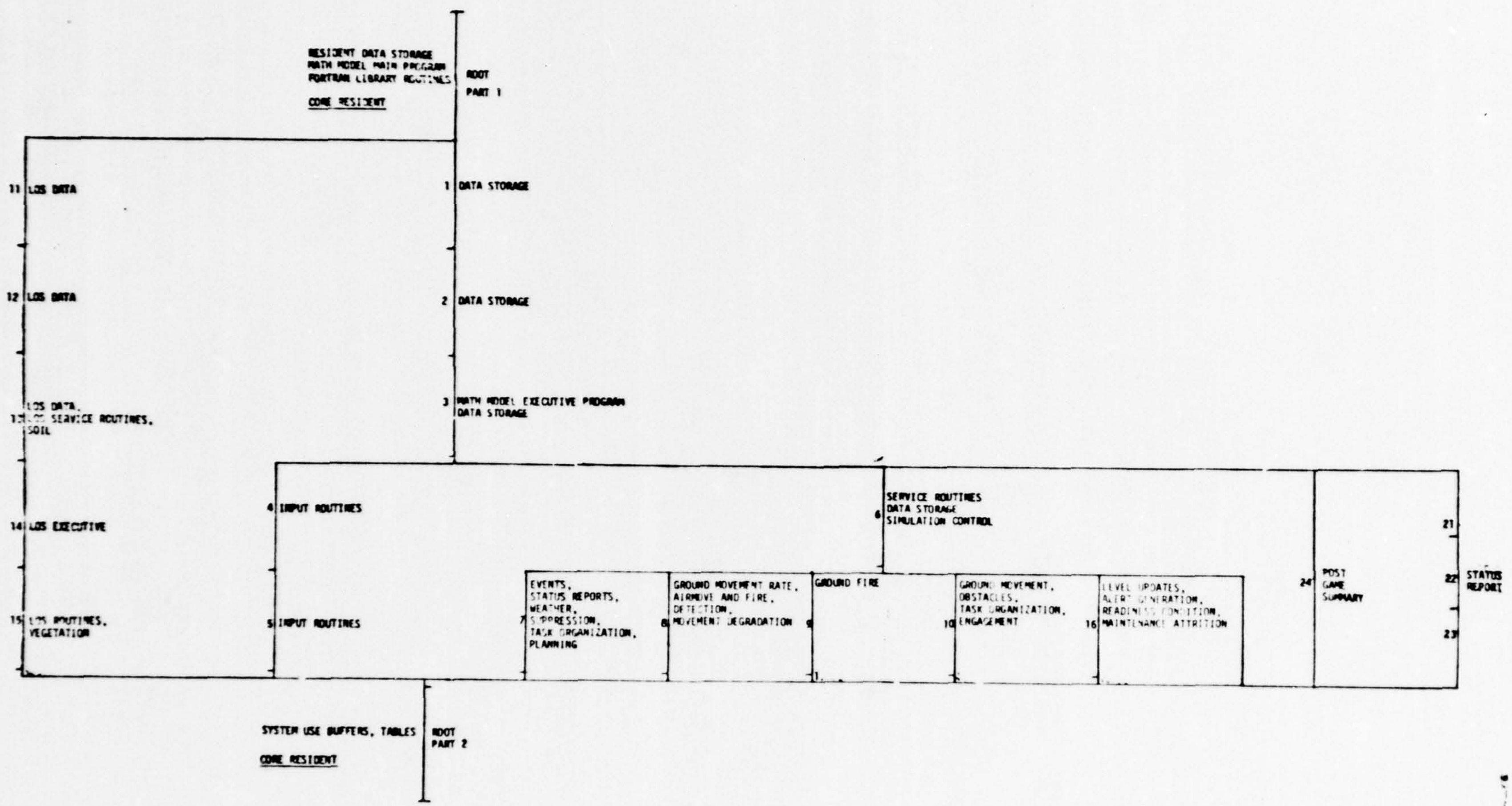


Figure 3-4 Overlay

3.103 PATH OF MOVEMENT

3.103.1 English-language Definition

a) Term Name:

Path of Movement

b) Term Description:

The path of movement for a given unit refers to the line segment determined by the unit's previous X-Y location and where it is scheduled to be relocated during the next time step. This new location is computed based on the knowledge of the unit's previous location, direction of movement, and movement speed. The unit will be relocated to this new location unless the following occurs:

1) the unit's ultimate destination will be reached somewhere within the line segment denoting the path of movement

2) the unit encounters an obstacle within the line segment denoting the path of movement.

The first case will cause the unit to be placed at its destination point, rather than at the new location computed. At this point, new movement instructions are given to the unit, and if time remains during the current time step, the unit is dispatched on its new move. The second case results in relocating the unit near the point where the unit encountered the obstacle. The unit is halted at this point and forced to endure the delay assessed against it.

c) Term References:

The path of movement is established every time step for each moving unit. The movement logic checks the unit's path of movement to determine destination arrivals and possible interaction with obstacles.

3.103.2 Programmatical Definition

(Not applicable)

3.104 PERSONNEL

3.104.1 English-language Definition

a) Term Name:

Personnel

b) Term Description:

The number of personnel in a unit is a key factor in the unit's operation. When a unit loses all its people, it becomes inactive. Personnel are lost primarily as a result of enemy fire, although they could be lost due to other factors. (e.g., minefields). Personnel casualties are primarily produced directly as a result of weapons firing, or indirectly as a result of equipment loss. As a unit's personnel are depleted, the unit's effectiveness is reduced in several ways. The unit may become more suppressed, in which case more personnel are placed in the suppressed mode of operation in order to become less vulnerably. A consequence, however, is that they become less effective. Personnel are distributed over equipments in order to man the equipments. Unmanned equipments have no effect on the unit's firing or movement capability. Therefore, as personnel are depleted, fewer equipments can be manned, and the unit's effectiveness is further reduced.

Personnel are also distributed over vulnerability class. See the term "personnel vulnerability class" for a discussion of its use. Personnel are also allocated to administrative categories solely for display purposes. These categories are:

1) commanding officer
2) officers
3) enlisted men-leaders
4) enlisted men

c) Term References:

Personnel is a unit attribute. It is used throughout the model to indirectly determine unit combat effectiveness, movement rate, and other major unit interactions. Unit personnel levels are depleted at the end of each time step after all firing has been completed.

3.104.2 Programmatical Definition

a) Term Name:

PERS(IU),MENNOW(J,IU)

b) Term Description:

PERS(IU) Number of personnel currently in unit IU.

MENNOW (J,IU) A halfword packed array which contains the current amount of the four types of personnel in each unit IU:

Left halfword (1,IU = No. of Co in unit IU

Right halfword (1,IU = No. of off in unit IU

Left halfword (2, IU = No. of EMLDR in unit IU

Right halfword (2,IU = No. of EM in unit IU

PERS(IU) and MENNOW(J,IU) are input initially as part of the unit definitions, and are normally updated once each time step after all casualty assessment has been completed. Personnel levels can also be chaged interactively via the RESUPPLY command and control action.

PERS and MENNOW are data base input variables, interactively generated variables, as well as model generated variables. As data base input variables, they are defined in the unit input deck; as interactively generated variables, tney are created from the resupply menu; and, as model generated variables, they are used by the Air, Ground Movement (Obstacle), and Ground Fire Modules.

c) Term References:

The above arrays are directly indexed by unit number, where ($1 \leq IU \leq 100$). A large number of subroutines use PERS and/or MENNOW, the most important of which are STEP and RESUPPLY. Subroutine STEP actually updates the PERS array to account for casualties sustained in the current minute. Subroutine RESUPPLY updates unit personnel levels as a result of a command and control action.

3.105 PERSONNEL VULNERABILITY CLASS

3.105.1 English-language Definition

a) Term Name:

Personnel Vulverability Class

b) Term Description:

The term personnel vulnerability class is used to describe the vulnerability of a man on the battlefield to the effects of different weapons, which is dependent largely on his posture (standing, hiding in a foxhole, etc.) and on how well he is schielded by equipment. Up to six different personnel vulnerability classes are allowed in the CATTS model, but only four are defined, as follows:

standing
prone
foxhole
invulnerable (completely shielded by an equipment)

Personnel in a unit are distributed among the four vulnerability classes each minute. First total personnel are distributed over equipment on a priority based on unit type (see "manning equipment" for details). Then the Ground fire module distributes manned equipment over modes of operation as a function mainly of unit operational state (for details, see discussion on "mode"). Each mode of operation of each equipment has a specific personnel vulnerability class associated with it. When all equipments have been processed by the Ground fire module, all personnel in the unit at the beginning of the time step will have been distributed over the four personnel vulnerability classes used. A unit's casualties are accumulated in a separate array. At the end of the time step, the casualties will be distributed over the classes according to a pre-defined weighting factor to reduce the levels of personnel in each class. These reduced levels will be used in the next time step in the allocation of fire against enemy units using another weighting factor to compute the value of the unit's personnel as a target. These reduced levels will also be reduced as casualties are incurred so that

people are not killed more than once. New personnel vulnerability class distributions will be computed next minute for that minute's actual personnel vulnerability class levels. Two sets of personnel vulnerability class levels are kept, therefore; one to represent the actual current time step's manning levels, the other to represent last time steps' manning levels after casualties are accounted for.

c) Term References:

The personnel vulnerability class associated with each mode is both equipment and a mode attribute. The personnel vulnerability class levels are primarily unit attribures. The other arrays help to define each personnel vulnerability class.

3.105.2 Programmatical Definition

a) Term Name:

IPVCE(IEQ,IMODE)
PERNVC(IU,IPVC,K)
WPVC(IPVC)
WTFCTR(IPVC)
CASTAB(JPVC)

b) Term Description:

The definitions of the program arrays related to the concept of personnel vulnerability class are as follows:

IPVCE (IEQ,IMODE)	For ground equipments (indicated by IEQCOD(IEQ) > = 0, personnel vulnerability class associated with each mode IMODE off equipment type IEQ. For air equipment other than an aircraft (IEQCOD (IEQ) = -2 or -3), equipment number of the IMODEth allowable aircraft type on which IEQ may be used. (Hence, a maximum of eight aircraft per equipment type.) The first zero encountered implies no other allowable aircraft.
PERNVC (IU,IPVC,K)	Number of personnel in each vulnerability class IPVC for unit IU where K=: 1-actual number in previous time step 2-actual number in current time step

	3-unsuppressed number in previous time step 4-unsuppressed number in current time step.
WPVC (IPVC)	Weight of a man as a target element for each personnel vulnerability class IPVC.
WTFCTR (IPVC)	Weighting factor used by subroutine step to distribute personnel casualties to vulnerability classes other than class 1, which is invulnerable (IPVC=1,5 for 6 possible classes)
CASTAB (JPVC)	Number of casualties in personnel vulnerability class JPVC of a unit (temporary)

IPVCE and WPVC are data base input variables that are defined in the equipment and first (miscellaneous variable) input decks, respectively.

PERNVC, WTFCTR, and CASTAB are model generated variables. PERNVC is used in subroutines ADW, AIRCAS, AMOVUL, CRDLIC (no menu was ever written which would cause this subroutine to be called), INIT, OBSDELAY, ORGFIR, SPTALO, STEP, WPNEFF, and WPNFIR. WTFCTR is used by subroutine FORMAIN (data statement), and CASTAB is used by subroutine WPNFIR.

The personnel vulnerability classes indicated by index IPVC are as follows:

1 = invulnerable
2 = foxhole
3 = prone
4 = standing
5 = not used
6 = not used

Therefore, IPVCE can only take on one of four values as presently defined in the data base, and only four of the six vectors or words in the arrays indexed by IPVC are used.

c) **Term References:**

Array IPVCE is indexed first by equipment number, IEQ($1 \leq$ IEQ ≤ 80, second by mode, IMODE($1 \leq$ IMODE ≤ 8). Array PERNVC(IU,IPVC,K) is indexed first by unit number, IU ($1 \leq$ IU ≤ 100), second by personnel vulnerability class, IPVC($1 \leq$ IPVC ≤ 6), although only 4 are used, and

finally by a dummy index, K($1 \leq K \leq 4$) defined above. The other arrays are indexed by personnel vulnerability class, IPVC ($1 \leq$ IPVC ≤ 6) only. Arrays PERNVC and CASTAB are calculated every time step; the others are input and never changed. Both PERNVC and CASTAB are accumulated in the Ground fire module, since all ground equipments are processed there. At the end of the time step, subroutine STEP reduces the personnel levels to account for casualties, and transfers the new personnel levels to the old personnel levels to prepare for the next time step.

3.106 PLATOON

3.106.1 English-language Definition

a) Term Name:

Platoon

b) Term Description:

A platoon is designated in the model by setting the size value for a specific unit to 3. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for platoon is "...".

The term platoon is also associated with the command and control panel button selections. When platoon is selected, all unit sizes having a designated value smaller than or equal to platoon are displayed on the appropriate menu.

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a size of 3 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 3 used to represent platoon.

3.106.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit input deck for units 1-100, the operational grouping input deck for units 101-120, and the highter and adjacent unit input deck for units 121-150.

A platoon is the designation for Unit I in the model by setting array ISIZE(I) to a value of 3 as part of the data base inputs. This value is never altered for units. A platoon is designated for operational

group (1-100) by setting array ISIZE(I),(101 $\leq$ I $\leq$ 120), equal to 3 to designate the appropriate size of the op group. This designation is mode via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 3 for ISIZE(I), where I is in the range (121 $\leq$ I $\leq$ 150), represents the designation of platoon for red and blue adjacent units represented [illegible] those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from (1 $\leq$ I $\leq$ 100) for normal units, (101 $\leq$ I $\leq$ 120) for op groups, and (121 $\leq$ I $\leq$ 150) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.107 PRE-PLANNED MISSION

3.107.1 English-language Definition

a) Term Name:

Pre-planned Mission

b) Term Description:

Pre-planned missions are pre-constructed packages of commands. The data required to specify a pre-planned mission is defined via input by the user. The input is read in and written onto the pre-planned missions file (on disk) in a direct access fashion. Each mission is identified by a unique name and consist of a number of event notices detailing the mission. The event notices are activated interactively by using the pre-planned missions menu. Upon activation, the event notices are read from the pre-planned missions file (on disk); the event notices are routed to the appropriate event processor. The processor modifies all required math model variables in order to implement the desired mission. It should be noted that a pre-planned mission events package cannot contain another pre-planned mission command (i.e., not a recursive procedure). Presently a maximum of ten pre-planned missions may exist in the model simultaneously.

c) Term References:

A pre-planned mission is referenced by its unique name when the pre-planned missions menu is displayed. Of course, the mission name and data, stored in the above arrays, are referenced by an integer (between 1 and 10 inclusively) identifying which of the ten possible missions is being selected. This integer accesses an a-ray element which will point to the location of the event notice data comprising the mission. Pre-planned missions are used by the command and control module to allow a controller to activate a series of pre-constructed (defined by input) event notices. This relieves the controller from having to interactively enter a number of individual event notices to achieve an objective.

3.107.2 Programmatical Definition

a) Term Name:

IPPREC(10),NAMEPLAN(8,10),NPPREC(10)

b) Term Description:

The integer arrays IPPREC, NAMEPLAN, and NPPREC contain data representing the attributes of pre-planned mission. Note that the arrays are dimensioned by ten indicating that storage is available for a maximum of ten missions at one time. The array NAMEPLAN stores the unique names associated with each mission. The array IPPREC contains pointers indicating the location of data describing the event notices of each mission. NPPREC contains the number of event notices comprising each pre-planned mission.

IPPREC(IPPM)	Starting disk record number for pre-planned mission no. IPPM
NAMEPLAN(2,IPPM)	Name associated with pre-plan mission no. IPPM. 8 characters each
NPPREC(IPPM)	Number of event notices (=no. of disk records) comprising pre-planned mission IPPM

NAMEPLAN and NPPREC are data base input variables that are defined in the pre-planned mission input deck. IPPREC is a model generated variable that is used in subroutine PPLANINP.

c) Term References:

The arrays used to define the attributes of pre-planned missions are indexed by IPPM, where IPPM is an integer ranging from 1 to 10 inclusively. These arrays are initialized by data base input; the data remains constant throughout the simulation. The following subroutines make reference to one or more of the above arrays:

FORRAM
PPLANINP
PREPLAN

3.108 PRE-SCHEDULED EVENTS

3.108.1 English-language Definition

a) Term Name:

Pre-scheduled Events

b) Term Description:

The CATTS math model provides two ways to input prescheduled events:

1) Events on the pre-scheduled events file.

2) Events in the card run deck used to run the model.

Obviously, to use method 2 (the card run deck), cards must be punched and placed in the proper order in the run deck. The order is as follows: The number of events must be namelisted in the namelist portion of the run deck, then immediately after the namelist section of the deck must come the events (includes both the 64 words of event notice data and the time of the event in namelist form) each followed by an asterisk.

The pre-scheduled events disk file may be filled using two methods:

1) By punching cards in the proper form and then doing a RADEDIT copy to get the card images on the disk file.

2) By using the procedure from the Data Base/Operations Manual for saving initializing conditions. This allows the events input via the menus during the three steps of initialization to be saved on the pre-scheduled events file.

The format for the pre-scheduled events file is as follows: First the number of events on the pre-scheduled events file (in namelist form) followed by an asterisk, then the events (includes both the time the event is to occur and the 64 words of event notice data in namelist form) each followed by an asterisk.

This capability to schedule events non-interactively has many possible uses. For example, the pre-scheduled events file can be used for events which occur every time a scenario is run such as weather changes, maneuvers, fire missions, etc. The card run deck events can be used

to test an event and for subsystem tests. All the events for a particular subsystem test could be punched on cards and inserted in the run deck for the particular subsystem test. This would ensure that the test would be exactly repeatable. Another potential use: An entire red battle plan could be input via the pre-scheduled event file or the card run deck, and the blue could thus be tested against a constant red battle plan.

3.108.2 Programmatical Definition

a) Term Name:

ITMEVE
IVDT(64)
NEVENT
NEVTP

b) Term Description:

ITMEVE — Time (in minutes since simulation start) at which event that is being scheduled will occur (on both pre-scheduled events file and run card deck events).

IVDT(64) — Event notice data of the event that is being scheduled (on both pre-scheduled events file and run card deck events).

NEVENT — The total number of events being input via the card run deck (if there are any events in the run card deck, this variable should be set in the namelist portion of the run deck).

NEVTP — The total number of events being input on the pre-scheduled events deck.

ITMEVE and IVPT(64) are data base input variables that are defined in the run deck and as card images on the pre-scheduled events file. NEVENT and NEVTP are data base input variables that are defined in the namelist portion of the run deck and as a card image at the front of the pre-scheduled events file, respectively.

c) Term References:

Both the events from the pre-scheduled events file and the run deck are read in by subroutine INPUT and then placed on the background events file by subroutine PEVENT. Then at the scheduled time for each of the events, the events module will call the proper subroutine and the event will be processed.

3.109 RAM ALERTS

3.109.1 English-language Definition

a) Term Name:

RAM Alerts

b) Term Description:

RAM alerts are CATTS alert messages that require immediate response by the operator. These are messages resulting from air missions, support fire commands, and simulation control commands. These alert messages are displayed immediately on to the television screens at the controller console by the foreground (interactive display) software. The controller, in turn, must react to them before he can use other menus.

c) Term References:

Table 3-3 is a list of the mathematical model subroutines that generate RAM alert messages.

3.109.2 Programmatical Definition

(Not applicable)

Table 3-3. Mathematical Model Subroutines That Generate Alert Messages

EVENTERR:

Type 1:	MANEUVER COMMAND CANCELLED AT time INVALID UNIT SELECTED
Condition:	Event notice unit designation is incorrect
Explanation:	Command is ignored/possible incorrect preplanned mission or possible programming error.
Type 2:	RESUPPLY COMMAND CANCELLED AT time INVALID UNIT SELECTED
Condition:	Event notice unit designation is incorrect
Explanation:	Command is ignored/possible incorrect preplanned mission or possible programming error

FORSIGI:

Type 3:	INITIALIZING FOR SCENARIO scenario name HIT IGNORE WITH GRAF-PEN ENTER ANY DESIRED COMMAND AND CONTROL WHEN COMPLETED, PRESS CONTROL SWITCH
Condition:	Completion of reading input file and namelist
Explanation:	None; operator may issue any desired commands; operator must press control switch in order for model to continue
Type 4:	LAST CHANCE TO RELOCATE HIT IGNORE WITH GRAF-PEN ENTER ANY DESIRED COMMAND AND CONTROL WHEN COMPLETED, PRESS CONTROL SWITCH
Condition:	Completion of processing; all commands entered after first initialization
Explanation:	None; if operator intends to relocate units he must do so now; operator must press control switch in order for model to continue

Type 5: MAGICAL MOVEMENT OF UNITS ALLOWED ONLY DURING INITIALIZATION --- REQUEST IGNORED FOR UNIT unit name

Condition: An attempt was made to relocate units during game

Explanation: Command causing the alert is ignored.

ACTIVATE:

Type 6: ONLY ONE DEACTIVATE UNITS COMMAND IS ALLOWED PER MINUTE HIT IGNORE TO CONTINUE

Condition: An attempt was made to perform unit deactivation twice for the same color, within the same minute

Explanation: Second deactivation command is ignored

CONTMS:

Type 7: CONTROL MEASURE control measure name NOT ADDED MAXIMUM NUMBER OF 100 ALREADY DEFINED
YOU MUST DELETE EXISTING MEASURE BEFORE ADDING NEW ONES

Condition: An attempt was made to add a new control measure when table was full

Explanation: Command causing alert is ignored; operator must delete an existing control measure before adding a new one

Type 8: CONTROL MEASURE control measure name NOT ADDED IMPROPER SPECIFICATION

Condition: Control measure subevent type is not 1 (add), 2 (delete), or 4 (change)

Explanation: Command causing alert is ignored

Type 9: NQPTOT 1 or 2/red or blue = value GREATER THAN 40.

Condition: Total number of red or blue firing equipment types (weapons) exceeds 38 (two columns of STATS array reserved for losses due to air and miscellaneous)

Explanation: Model continues processing; indices to array STATS may go out of range causing unpredictable results and possible job abortion

Type 10: Event name COMMAND FOR UNIT unit name IS LOST DUE TO DEAD COMMANDING OFFICER

Condition: An attempt was made to issue a command to a unit having lost communication due to destruction of command post

Explanation: Command causing alert is ignored

LEAVEOG:

Type 11: OP GRP op grp name NOW DEFUNCT

Condition: The operational group specified in a "leave OG" command is defunct

Explanation: Command causing alert is ignored

MANEUVER:

Type 12: unit or op. grp. name MANEUVER COMMAND CANNOT BE TAKEN DUE TO IMPROPER SPECIFICATION.

Condition: Event notice command contains erroneous data

Explanation: Command causing alert is ignored

Type 13: unit or op. grp. name MANEUVER COMMAND CANNOT BE TAKEN DUE TO MAXIMUM SP ROUTES REACHED

Condition: The maximum number of special routes has been exceeded

Explanation: Command causing alert is ignored

Type 14: unit or op. grp. name MANEUVER COMMAND CANNOT BE TAKEN DUE TO UNIT SUPPRESSION > 90%

Condition: Self-explanatory

Explanation: Command causing alert is ignored

Type 15: ORDERS FOR OG op. grp. name IGNORED INITIALIZATION MUST BE COMPLETED BEFORE OG MANEUVER CONTROL

Condition: A maneuver control command was issued prior to completion of initialization

Explanation: Command causing alert is ignored

NEWFWDUN:

Type 16: OP GRP op grp name NOW DEFUNCT

Condition: An existing operational group has disbanded

Explanation: Model continues with the units forming the disbanded op group treated as individual units

PREPLAN:

Type 17: NONEXISTENT PREPLANNED MISSION

Condition: Information in the event notice received by the model contained an index to a non-existent preplanned mission

Explanation: Command causing alert is ignored

RAMGEN:

Type 18: FIRE COMMAND REJECTED unit name equipment name NO SUCH WPN IN UNIT

Condition: Weapon type in event notice does not exist in unit specified

Explanation: Command causing alert is ignored

Type 19: FIRE COMMAND REJECTED unit name equipment name FIREVENT TABLE FULL

Condition: Self-explanatory

Explanation: Command causing alert is ignored

Type 20: FIRE COMMAND REJECTED unit name equipment name TOO MANY ROUNDS REQUIRED

Condition: The weapon designated in a Rounds Per Minute Fire command cannot fire the total number of rounds requested

Explanation: Command causing alert is ignored

Type 21: FIRE COMMAND REJECTED unit name equipment name DIRECT FIRE/AREA TARGET

Condition: A weapon other then artillery was commanded to fire at a bridge, road, or XY point

Explanation: Command causing alert is ignored

Type 22: FIRE COMMAND REJECTED unit name equipment name
RPM CAN'T OVERRIDE PCT

Condition: A unit/weapon pair having an existing percent of fire command was commanded to fire a specified number of rounds per minute

Explanation: Command causing alert is ignored

Type 23: FIRE COMMAND REJECTED unit name equipment name
NONFIRING EQUIPMENT

Condition: Event notice received by math model specified a non-firing equipment

Explanation: Command causing alert is ignored

Type 24: FIRE COMMAND REJECTED unit name equipment name
TARGET OUT OF RANGE unit name

Condition: Target unit specified in fire command is out of range

Explanation: Command causing alert is ignored

Type 25: FIRE COMMAND REJECTED unit name equipment name
TARGET OUT OF RANGE bridge or road or x y point
X = x loc. Y = y loc.

Condition: Target (bridge, road, or point) specified in fire command is out of range

Explanation: Command causing alert is ignored

AIRERROR:

Type 26: AIR MISSION air unit name ERROR error #
INVALID EVENT SUBTYPE
COMMAND IGNORED/CANCELLED

Condition: Self-explanatory

Explanation: Self-explanatory

Type 27: AIR MISSION air unit name ERROR error #
INVALID MISSION TYPE
COMMAND IGNORED/CANCELLED

Condition: Self-explanatory

Explanation: Self-explanatory

Type 28: AIR MISSION air unit name ERROR error #
NO AVAILABLE UNITS
COMMAND IGNORED/CANCELLED

Condition: An attempt was made to create an air unit when 10 air units (maximum allowed) already exist

Explanation: Self-explanatory

Type 29: AIR MISSION air unit name ERROR error #
INVALID AIRCRAFT TYPE
COMMAND IGNORED/CANCELLED

Condition: Self-explanatory

Explanation: Self-explanatory

Type 30: AIR MISSION air unit name ERROR error #
INVALID EQUIPMENT DELETED equipment name
MISSION CONTINUED

Condition: Specified aircraft is not able to carry equipment in alert message

Explanation: Air unit is created without the disallowed equipment

Type 31: AIR MISSION air unit name ERROR error #
LOAD EXCEEDED - DELETED EQ equipment name
MISSION CONTINUED

Condition: Specified equipment caused aircraft load capacity to be exceeded

Explanation: Air unit is created without the disallowed equipment, if possible

Type 32: AIR MISSION air unit name ERROR error #
INVALID OLD UNIT SELECTED
MISSION CONTINUED

Condition: A change command involved a non-existent air unit

Explanation: Self-explanatory

Type 33: AIR MISSION air unit name ERROR error #
BAD WEATHER
COMMAND IGNORED/CANCELLED

Condition: Meteorological visibility is too low for that specified for the aircraft

Explanation: Self-Explanatory

Type 34: AIR MISSION air unit name ERROR error #
FUEL REQUIRED EXCEEDS CAPACITY
COMMAND IGNORED/CANCELLED

Condition: Air craft cannot carry enough fuel for the designated route and load.

Explanation: Self-explanatory

Type 35: AIR MISSION air unit name ERROR error #
INVALID NUMBER OF ROUTE POINTS
COMMAND IGNORED/CANCELLED

Condition: Less than four or more than nine route points were specified for the air unit

Explanation: Self-explanatory

Type 36: AIR MISSION air unit name ERROR error #
OLD MISSION NOT YET IMPLEMENTED
COMMAND IGNORED/CANCELLED

Condition: An attempt was made to change a previously created air mission that has not taken off

Explanation: The attempt to change the mission is ignored; original air mission is left intact

Type 37: AIR MISSION air unit name ERROR error #
INVALID TARGET
COMMAND IGNORED/CANCELLED

Condition: Event notice contains invalid target number, e.g., defunct unit

Explanation: Self-explanatory

Type 38: AIR MISSION air unit name ERROR error #
AIRCRAFT PERFORMANCE EXCEEDED - LIMIT USED
MISSION CONTINUED

Condition: Aircraft speed or altitude maximum or minimum exceeded

Explanation: Self-explanatory

Type 39: AIR MISSION air unit name ERROR error #
MISSION ALREADY CANCELLED
COMMAND IGNORED/CANCELLED

Condition: An attempt was made to cancel a previously cancelled mission

Explanation: Self-explanatory

Type 40: AIR MISSION air unit name ERROR error #
DENSITY ALTITUDE ABOVE AIRCRAFT CAPABILITY
COMMAND IGNORED/CANCELLED

Condition: A newly created air unit is not able to take off due to high density altitude

Explanation: Air mission is ignored

Type 41: AIR MISSION air unit name ERROR error #
NO ORDNANCE AVAILABLE FOR THIS TARGET
COMMAND IGNORED/CANCELLED

Condition: An air strike was attempted against a target, but no valid ordnance was given to the aircraft to perform the strike

Explanation: Air mission is ignored

Type 42: AIR MISSION air unit name ERROR error #
MIN SPEED TOO LOW FOR EQUIP -- MIN USED
MISSION CONTINUED

Condition: The velocity specified for an aircraft is too low for requested equipment

Explanation: Minimum velocity for ordnance delivery is used; mission continued

Type 43: AIR MISSION air unit name ERROR error #
MAX SPEED TOO HIGH FOR EQUIP -- MAX USED
MISSION CONTINUED

Condition: The velocity specified for an aircraft is too high for requested equipment

Explanation: Maximum velocity for ordnance delivery is used; mission continued

Type 44: AIR MISSION air unit name ERROR error #
MIN ALT TOO LOW FOR EQUIP -- MIN USED
MISSION CONTINUED

Condition: The altitude specified for an aircraft is too low for requested equipment

Explanation: Minimum altitude used; mission continued

Type 45: AIR MISSION air unit name ERROR error #
MAX ALT. TOO HIGH FOR EQUIP -- MAX USED
MISSION CONTINUED

Condition: The altitude specified for an aircraft is too high for requested equipment

Explanation: Maximum altitude used; mission continued

TASKORG:

Type 46: NO ROOM FOR NEW OP GROUP

Condition: OP GRP table is full

Explanation: Command causing alert is ignored

Type 47: IN TASKORG ERROR # error #

Condition: Error # cause

1. No unit in the event notice
2. Event notice sub-type not a legal value
3. Size of units given by event notice is incorrect
4. Op group number given by event notice is incorrect
5. Unit doesn't belong to the OG you are trying to delete it from
6. Trying to add a unit to the OG it already belongs to
7. Can't determine color of OG in create command
8. Less than 2 units in an attempted creation of new OG

Explanation: Command causing alert is ignored

TESTSAVE:

Type 48: SAVE FILE RECORD LENGTH TOO SMALL
REPLAY AND RESTART CAPABILITY DELETED

Condition: The allocated size of the save file is too small

Explanation: Subsequent attempts to replay or restart will be ignored

Type 49: NO SAVE FILE SPECIFIED
REPLAY AND RESTART CAPABILITY DELETED

Condition: No save file was allotted as part of load or run deck

Explanation: Subsequent attempts to replay or restart will be ignored

Type 50: INITIALIZATION COMPLETE
HIT IGNORE WITH GRAF-PEN
PRESS CONTROL SWITCH TO START SIMULATION

Condition: Initialization load of all model variables completed, including line of sight calculations

Explanation: Model halts processing; operator must press control switch to resume model processing

Type 51: REPLAY WILL START AT day # DAY, hour # HOUR, min # MINUTE
PRESS CONTROL SWITCH TO COMMENCE REPLAY

Condition: A replay has been requested to begin at a specified time

Explanation: Model halts processing; operator must press control switch to initiate replay

Type 52: REPLAY TERMINATED
SIMULATION CONTINUING AT day # DAY, hour # HOUR, min # MINUTE

Condition: A replay was terminated before the end is reached

Explanation: Model continues processing at point before replay

Type 53: REPLAY COMPLETED
SIMULATION CONTINUING AT day # DAY, hour # HOUR, min # MINUTE

Condition: A replay has just completed

Explanation: Model continues processing from end of replay at specified time

Type 54: SAVE FILE FULL
GAME TERMINATING

Condition: No more data can be saved for replay because the save file is full

Explanation: Model terminates

Type 55: UNABLE TO INITIALIZE NEW SCENARIO
CONTINUING WITH OLD SCENARIO

Condition: Attempt was made to reinitialize a nonexistent scenario

Explanation: Model continues processing with current scenario

Type 56: REPLAY/RESTART CAPABILITY HAS BEEN DELETED
REQUEST IGNORED
SIMULATION CONTINUING

Condition: Replay/Restart capability has been deleted due to inadequate allocation of save files

Explanation: Self-explanatory

Type 57: SIMULATION BEING RESET TO day # DAY, hour # HOUR, min # MINUTE
PRESS CONTROL SWITCH TO COMMENCE RESTART

Condition: Restart has been requested at specified time

Explanation: Model halts processing; operator must press control switch to initiate restart

Type 58: ILLEGAL TIME SPECIFIED
REQUEST IGNORED
SIMULATION CONTINUING

Condition: An illegal time non-existent on the save files has been requested for replay/restart

Explanation: Self-explanatory

3.110 RANDOM NUMBER GENERATOR

3.110.1 English-language Definition

a) Term Name:

Random Number Generator

b) Term Description:

In the CATTS mathematical model, a number of the important tactical and physical mathematical relationships are represented by a stochastic process with probability distributions or parameters that satisfy the required mathematical properties of the equations. The generation of the associated statistics (random variates or numbers) is entirely numberical in nature and is carried out by supplying pseudorandom numbers, generated on the computer, into a probabilistic model and obtaining random numbers from it as answers.

c) Term References:

The random number generator is used to generate sequences of random variates from probability distributions used in the CATTS mathematical model.

3.110.2 Programmatical Definition

a) Term Name:

RANDU(IRAND)

b) Term Description:

RANDU(IRAND) is a library function that causes the variable RN to be set equal to a pseudorandom number generated by the function RANDU. The initial value of IRAND is an odd integer or "seed" and is stored in the common block RANDOM.

c) Term References:

The following subroutines use RANDU:

ADW	GRNDAIR
AIRCAS	NORMAL
AIRGRND	OBSDELAY
AIRHOTZN	STEP
DETECT	WETHR
DMGEVAL	WPNEFF
EFFNS	

3.111 RATE OF FIRE

3.111.1 English-language Definition

a) Term Name:

Rate of Fire

b) Term Description:

Each firing weapon in the model has an associated firing rate. Any equipment whose code is greater than zero is a ground fire weapon. Storage allows for up to eight different firing rates for a weapon. The present data base configuration, in conjunction with the assumed mode definitions (see the definition for the term "mode") utilize two firing rates for direct, indirect and air defense fire weapons (equipment Codes 1, 2, and 4). Mode 6 is assigned the sustained firing rate of the weapon. Mode 7 is assigned the maximum firing rate. All other modes have a firing rate of zero. For support fire weapons (equipment code 3), six different firing rates are defined as follows:

MODE	FIRING RATE
3	Close support, normal
4	Close support, final protective fire
5	Interdictory
6	General support fire against op groups
7	Counterbattery
8	General support fire against individual units

The number of rounds fired from a weapon is a particular unit is computed using the mode distribution vector, which allocates fractions of the total number of weapons to each of the eight modes. The number of weapons in each mode is multiplied by the firing rate of that mode, then all modes are accumulated to yield the total number of rounds.

c) Term References:

The ground fire function uses the fire rates to determine the overall firepower of each ground weapon in a unit. The data is input at initialization time for each of the eight modes, for each of the 80 equipments, and never changed.

The Air modules utilizes the aircraft fuel performance data and the ordnance lethality fractions. This data is also input at initialization.

3.111.2 Programmatical Definition

a) Term Name:

ROFE(IEQ,IMODE)

b) Term Description:

The array ROFE is a data base input variable that is defined in the equipment input deck. It represents fire rates for ground fire weapons (IEQCOD(IEQ) ≥ 0), aircraft fuel performance data for aircraft (IEQCOD(IEQ) = -1), and lethality fractions for air ordnance (IEQCOD(IEQ) = -2). It is not used for air sensors or non-firing ground equipment.

The various definitions are as follows:

ROFE (IEQ,IMODE) For ground equipments (indicated by IEQCOD(IEQ) > = 0), rate of fire of weapon type IEQ in each mode IMODE (rounds/minute)

For aircraft (IEQCOD(IEQ) = -1),

IMODE = 1 fuel expenditure for losing altitude

2 fuel expenditure for gaining altitude (lb/meter)

3 fuel expenditure at minimum speed, minimum load, best pressure density (lb/min)

4 fuel expenditure at cruise speed (lb/min)

5 fuel expenditure at maximum speed (lb/min)

6 ratio of fuel expenditure rate at maximum load to fuel expenditure rate at minimum load

7 ratio of fuel expenditure rate at worst pressure density to fuel expenditure rate at best pressure density

8 not used

For air ordnance (IEQCOD(IEQ) = -2),

IMODE = 1 fraction of personnel in personnel vulnerability class 1 (standing) and within target area who are killed by this equipment

2 fraction of personnel in personnel vulnerability class 2 (crouching) and within target area who are killed by this equipment

3 fraction of personnel in personnel vulnerability class 3 (prone) and within target area who are killed by this equipment

4 fraction of equipment with IEQCLS = 1 and within target area which is damaged by this equipment

5 fraction of equipment with IEQCLS - 2 and within target area which is damaged by this equipment

6 fraction of equipment with IEQCLS = 3 and within target area which is damaged by this equipment

7 fraction of equipment with IEQCLS = 4 and within target area which is damaged by this equipment

8 fraction of equipment with IEQCLS = 5 and within target area which is damaged by this equipment

c) Term References:

Array ROFE(IEQ,IMODE) is referenced primarily by equipment number, IEQ, ranging between ($1 \leq$ IEQ ≤ 80). The secondary index is mode number, where ($1 \leq$ IMODE ≤ 8). For aircraft and air ordnance (IEQCOD (IEQ) = -1 or 2), the second index is merely a dummy index with the range one through 8.

The following ground fire subroutines are principal users of ROFE:

AIRCAS
AMOVUL
FIRALO
ORGFIR
SPTALO

The following Air module subroutines are principal users of ROFE:

ADW
AIREVENT
AIRMOV
DIDITHIT

3.112 READINESS CONDITION

3.112.1 English-language Definition

a) Term Name:

Readiness Condition

b) Term Description:

Readiness condition is calculated once per time step for each active simulated unit. It is calculated according to the following three criteria:

1) Fraction of personnel basic load remaining
2) Fraction of number of pieces of equipment in the basic load remaining
3) Number of equipment types currently falling into each readiness category -- based on fraction of unit basic load remaining for that type.

The numerical thresholds used were taken from USAIS pamphlet. An alert message is generated whenever any unit changes readiness condition.

c) Term References:

Readiness condition is a ground unit attribute; not used for air units.

3.112.2 Programmatical Definition

a) Term Name:

IREDCON(IU)

b) Term Description:

IREDCON is a model generated variable that is used in subroutines REDCON and CRDLIC (no menu was ever written which would cause subroutine CRDLIC to be called).

The definition for IREDCON is as follows:

IREDCON (I) The readiness condition for Unit I

IREDCON is an integer between 1 and 4 inclusive describing unit readiness condition. It is calculated in REDCON once per time step.

c) **Term References:**

IREDCON(IU) is indexed on unit number, IU($1 \leq IU \leq 100$). Subroutine REDCON sets the array, and the line printer and Superbee output routines principally use it.

3.113 RECOVERY

3.113.1 English-language Definition

a) Term Name:

Recovery

b) Term Description:

There are a number of possible occurrences which can cause an involuntary abnormal termination of a CATTS exercise. These include hardware failure, software failure, and data base errors. While every attempt is made to anticipate and prevent such occurrences, they were and are an inevitable part of the operation of a large, complex system. Because such failures, though rare, can impose substantial inconvenience and loss of training effectiveness, the CATTS simulation control software was modified to automate "recovery" of the exercise to the restart point (which represents the last time the data base was saved on disk files) immediately preceding the failure. Thus, the exercise can usually be continued after only a few minutes interruption, rather than terminated or begun over again.

c) Term References:

None

3.113.2 Programmatical Definition

a) Term Name:

IRECOVER

b) Term Description:

IRECOVER - The number of minutes past game start time to which recovery is to be made.

IRECOVER is a data base input variable that is defined in the namelist portion of the card run deck.

c) Term References:

Recovery is accomplished in subroutine INPUT.

3.114 REGIMENT

3.114.1 English-language Definition

a) Term Name:

Regiment

b) Term Description:

A regiment is designated in the model by setting the size value for a specific unit to 6. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for regiment is "111".

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a size of 6 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 6 used to represent regiment.

3.114.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit input deck for units 1-100, the operational grouping input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150.

A regiment is designated for Unit I in the model by setting array ISIZE(I) to a value of 6 as part of the data base inputs. This value is never altered for units. A regiment is designated for operational group (I-100) by setting array ISIZE(I),(101 $\leq$ I $\leq$ 120), equal to 6 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task

organization menu. A value of 6 for ISIZE(I), where I is in the range ($121 \leq I \leq 150$), represents the designation of regiment for red and blue adjacent units represented by those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.115 REINITIALIZE

3.115.1 English-language Definition

a) Term Name:

Reinitialize

b) Term Description:

The principal controller is provided with the option of switching to a different scenario or resetting the current simulation back to the beginning. This "reinitialization" has the effect of restarting the current game or selected alternative scenario from the beginning. The principal controller selects the reinitialization option and desired scenario from the simulation control menu which is presented to him when he selects the simulation control switch on his display panel.

c) Term References:

The principal controller institutes the reinitilization procedure.

3.115.2 Programmatical Definition

a) Term Name:

There is no programmatical term name.

b) Term Description:

The reinitialization request is passed from the command and control system to the math model via a CAL4 executed in the math model. When such a request is received, the DCB's (Data Control Blocks) F:10, F:11 and F:12 are reassigned to the appropriate files after closing the existing ones. The main program of the math model is then called and executed. The simulation initialization process is reactivated at this point. Test scenarios or "non-standard" scenarios are handled differently from standard scenarios. The non-standard scenarios are scenarios that do not have their disk files defined in the main program of the mathematical model. The math model uses the file identification (disk area name and file-name) fo the binary data base (F;10) as the

identification of a scenario. During the initialization of the mathematical model, the inital assignment of F:10 is compared with the F:10 assignments of the standard scenarios whcih are DATA'd in the main program. If it is not one of the standard scenarios, a scenario name is formed by appending two asterisks in front of the F:10 file identification. This temporary scenario name along with the file identifications of F:10, F:11 and F:12 are stored as a scenario if there is room in the scenario definition arrays. This temporary scenario definition is not saved in the binary data base and thus cannot be reinitialized after the principle instructor reinitializes to one of the standard scenarios.

A few points should be considered when using the reinitialize command. Sense switch 3 on the computer panel is considered off regardless of its actual setting so that the reinitilization will be done using the binary file of the data base. (Sense switch 3 on normally would cause initialization from the card image source copy of the data base). The NAMELIST input option for updating the binary data base (ISAVEINP = 1) is ignored and set to no-update, which prevents the obviously redundant creation of a binary data base from the same binary data base just used for initialization. It should be realized that the simulator cannot determine whether the binary data base used to reinitialized is up to date or not. In fact, the simulator cannot always tell if the binary data base is initialized properly. The instructors, however, should be able to recognize an improper reinitialization from errors on the menus or the graphics displayed on the TV monitors.

c) Term References:

None

3.116 RELIEF

3.116.1 English-language Definition

a) Term Name:

Relief

b) Term Description:

Relief, from the standpoint of mathematical modeling, is considered to be the most important factor in terrain representation. That is, once the relief factor of terrain is modeled, the other factors of terrain (i.e., vegatation, soil, obstacles) will be modeled with the consideration of achieving a high degree of mathematical compatibility. The basic approach for treating relief is to divide the entire area of operation into grid squares. At each grid corner, elevation data of the ground is known. The size of the grid square (called grid resolution) is a user input which depends upon the availability of grid elevation data. Presently, military terrain tapes (from the Defense Mapping Agency) containing digitized elevation data at twenty-five meter resolution is available for use in the CATTS model.

An interpolation is conducted to approximate surface elevation at non-grid points. Specifically, the elevation at nongrid point is determined as a weighted average of the four corner elevations of the grid square in which the point lies. This gives rise to a continuous surface representation of relief yielding accurate results without significantly increasing computer run time.

c) Term References:

An enormous amount of data is required to model the terrain feature of relief in the area of operation. All relief data will not fit into the computer at one time. This requires the data to be divided into 64 equal blocks. Each relief data block is further subdivided into smaller squares, 25.4 meters on a side. In the X direction, the relief block will contain 472 smaller squares, and in the Y direction, 132. Because of the data packing scheme, the data for one block requires approximately 16000 words of memory, which is approximately the maximum number which can be read into the XEROX computer at one time. Relief is referenced only by the environmental/terrain module. All reference

and calculations with relief data is accomplished at the beginning of each time step. Computed results are stored away and referenced as needed by the math model during the remainder of the time step. Thus to save time, each of the 64 relief blocks is read into the computer once, and only once. All calculations (i.e., lines of sight, elevation and slope at unit locations) with a given block should be accomplished while the block is in core.

3.116.2 Programmatical Definition

a) Term Name:

IELEVE(34,472),IELEVO(34,472),IEVOSP(3,750)

b) Term Description:

Relief data is read by blocks into the above arrays. There are 64 such blocks, and each is read once per time step.

IELEVE (I,J)	Buffer area for reading elevation data in meters for a grid block. Data is byte packed for grid cells of dimensions DELYBS. The Jth column of the array contains elevation data for the line X=(J-1)DELXBS + XZERO, where XZERO is the X-coord. of the origin of the terrain data grid block. The first half word is a base elevation in meters for the column; the second halfword is 0 or a pointer to a block in the table IEVDSP used for col. along which the elevation changes by more than 255 meters; the next 132 bytes are increments in meters to be added to the base elevation (+ 256 x dispersion from IEVDSP when the pointer is NONZERO) to yield the total elev. at Y=(K-1) DELYBS + YZERO where YZERO is the Y-coord. of the origin of the terrain data grid block and $1 \leq K \leq 132$. The range of K is obtained by 4xI.
IELEVO (I,J)	Same as IELEVE. It is used to facilitate asynchronous I/O.

IEVDSO (I,J) Dispersion table for use with IELEVE or IELEVO for columns of a terrain block for which the elevation changes by more than 255 meters. The table is byte packed in 12 byte blocks. Each block consists of at most six pairs of row number and number of 256 meter increments to be added to the base elevation in IELEVE or IELEVO for all rows equal to or beyond the row number. The block of bytes is terminated by an 0 Byte when fewer than six pairs are required.

IELEVE, IELEVO, and IEVOSP are data base input variables. IELEVE and IELEVO are defined on the data base disk file RELIEF in the DA area of the disk. IEVOSP is defined on the data base disk file ELEVDISP in the DA area of the disk.

c) Term References:

The arrays IELEVE(M,N) and IELEVO(M,N) are indexed such that M($1 \leq M \leq 34$) and N($1 \leq N \leq 472$) reference the elevation data for the point defined by the intersection of the coordinate line X = (N-1)*DELXBS + XZERO and the coordinate line Y = (K-1)*DELYBS+YZERO, where K($1 \leq K \leq 132$) references the last 132 bytes provided by the M($1 \leq M \leq 34$) words indicated by the first subscript. XZERO,YZERO are the X-Y coordinates of the origin of the relief data block.

3.117 REPLAY

3.117.1 English-language Definition

a) Term Name:

Replay

b) Term Description:

The principle instructor at any desired point during the CATTS simulation may order a replay of the events that have just happened. He may further select the speed of the replay and during the replay he may, at any time he wishes, terminate the replay and continue with the simulation or enter another simulation command, including another replay command. During the replay process, instructors may select any combination of graphic display options regardless of the options selected while the game was being played. When the replay is terminated and the processing continues, the state of the simulator is restored to the state before the replay and the simulation is in no way affected.

c) Term References:

None

3.117.2 Programmatical Definition

(Not applicable)

3.118 RESTART

3.118.1 English-language Definition

a) Term Name:

Restart

b) Term Description:

During the course of the CATTS simulation, the principal instructor has the option of resetting the simulation to any selected restart point. These restart points are spaced at regular intervals selected as input. Currently, restart points are spaced 15 minutes of simulation apart. At the restart points, the state of the simulation is saved. When the restart command is given, the state of the simulation is restored to the restart point at the selected time or to the point closest to and before the selected time.

c) Term References:

None

3.118.2 Programmatical Definition

(Not applicable)

3.119 RESUPPLY

3.119.1 English-language Definition

a) Term Name:

Resupply

b) Term Description:

In the CATTS system, the term "resupply" refers to an event notice which provides for the resupply of a unit with up to fourteen equipment types, fourteen ammunition types, and all types of personnal and fuel. Each resupply event notice contains:

1) Identification of donor and recipient units.

2) The quantity of personnel in each of the four categories (commanding officer, officers, enlisted men in leadership positions, and enlisted men) to be transferred between the two units.

3) The identification and quantity of each ammunition type to be transferred.

4) The identification and quantity of each equipment type to be transferred.

5) The quantity of each fuel type (gasoline, diesel oil, and aviation gasoline) to be transferred.

The transfer of supplies between units (requpply) is based on the following considerations:

1) At least one of the units must be one of the 99 modeled units or no transfer takes place.

2) If either unit is one of the 99 units, then it must be currently active or no transfer takes place.

3) If the donor unit is one of the 99 units, then the quantities of items resupplied are set to the minimum of the current levels for the donor and the quantities requested.

4) When vehicles are resupplied, the current fuel load of the vehicle is also resupplied. When the donor unit is not one of the 99 units, then the maximum fuel load of the vehicle is resupplied.

5) If the recipient unit is one of the 99 units, the total number of equipment and/or ammunition types carried by that unit cannot exceed the maximum of 14 types. Therefore, whenever the number of types carried by the recipient unit reaches 14, no more new types may be added. Requests to do so are ignored.

6) Some of the firing logic makes assumptions regarding the ordering of the equipment lists for each unit. For this reason, a new equipment type resupplied into one of the 99 units may be inserted in the middle of its equipment list rather than at the end.

c) Term References:

None

3.119.2 <u>Programmatical Definition</u>

(Not applicable)

3.120 ROAD

3.120.1 English-language Definition

a) Term Name:

Road

b) Term Description:

A road is represented by a series of connecting line segments (called road segments). It is defined by the following attributes:

1) the number of points (X-Y coordinates) required to form the series of connecting line segments
2) the start point of the series of line segments
3) the type of road

The set of points (X-Y coordinates) forming a series of connecting line segments to model a road, is an ordered set. Thus a road having N points is uniquely defined by specifying the start point and connecting this point with the next N-1 points in succession with straight lines. A road type is assigned to distinguish the three kinds of road currently being modeled:

1) multi-lane, all weather, high speed road
2) single-lane, all weather, high speed road
3) single-lane, all weather, low speed road

Roads must be pre-defined via input. A maximum of fifty roads can exist simultaneously.

c) Term References:

The data defining roads is referenced by the road number (there can be a maximum of fifty roads defined, each identified by an integer ranging from one to fifty). Roads are utilized mainly by the movement function and the ground and air firing functions. The movement logic generally allows units to travel faster on roads. The firing functions attempt to damage roads to impede the progress of oncoming enemy forces.

3.120.2 Programmatical Definition

a) Term Name:

IRDNPTS(50),IRDSTRT(50),IRDTYPE(50)

b) Term Description:

The above arrays contain integer data which will define all roads being modeled. The arrays IRDNPT and IRDSTRT reference arrays containing X-Y coordinate values of points defining each segment of the road. The array IRDTYPE contains a code for each road characterizing the type of road being modeled. The definitions of the above arrays may be summarized as follows:

IRDNPTS(IRD = the number of successive points in the IRDth ($1 \leq IRD \leq 50$) road; this number ranges from two to possibly five hundred

IRDSTRT(IRD) = The number of the first X-Y coordinate point in arrays IROADX and IROADY which belongs to the IRDth ($1 \leq IRD \leq 50$) road

IRDTYPE(IRD) = the number of the first X-Y coordinate point in arrays IROADX and IROADY which belongs to the IRDth ($1 \leq IRD \leq 50$) road

IRDTYPE(IRD) = type of the IRDth ($1 \leq IRD \leq 50$) road, where:

1 = multi-lane all weather high speed
2 = single-lane all weather high speed
3 = single-lane all weather low speed

IRDNPTS and IRDTYPE are data base input variables that are defined in the road input deck. IRDSTRT is a model generated variable that is used in subroutine ROADINP.

c) Term References:

The above arrays are indexed by IRD, the road number, where IRD is an integer between one and fifty inclusively.

3.121 ROAD SEGMENTS

3.121.1 English-language Definition

a) Term Name:

Road Segments

b) Term Description:

Road segments are individual entities which comprise a road. Each road segment is defined to be the line segment modeling the portion of road between two successive points (i.e., X-Y coordinates) describing the road. Thus an attribute of road segment are the X coordinates and the Y coordinates of the successive endpoints of the line segment. A road is a set of connecting road segments.

Another attribute associated with each road segment is damage assessed against it due to air or ground ordnance. The damage is assumed to be uniformly distributed over the entire road segment and acts to retard the movement of units attempting to travel over it.

c) Term References:

Road segment for a particular road is referenced relative to the start point of that road. In other words, in order to examine the Kth segment of a given road having a start point at the Jth endpoint, the (J+K-1)th and (J+K)th endpoint must be referenced.

Road segments comprise roads, hence all model functions which deal with roads deal also with road segments. This would include the movement function which is concerned with overall unit movement rate along roads, and the air and ground ordnance functions which inflict damage upon road segments.

3.121.2 Programmatical Definition

a) Term Name:

IROADX(500),IROADY(500),RDDMGE(500)

b) Term Description:

The X-Y coordinate values of endpoints of all road segments are stored in the integer arrays IROADX and IROADY. Roads are defined by specifying the number of successive points and the start point of the road. Specifically to define road N, the integer stored in the element IRDSTRT(N) indicates the initial point of the road, and the integer stored in the element IRDNPTS(N) indicates the total number of successive points to form the required road segments. (See definition of road).

The arrays IROADX and IROADY contain integers which describe coordinate values within the area of operation. The floating point array RDDMGE contains fractions ranging from zero to one inclusively describing damage to each segment. A value of zero indicates no damage and a value of one indicates total damage.

The following summarizes the representation of road segments, where $1 \leq \text{IRDSG} \leq 500$ and $1 \leq N$ _ 50:

IROADX (IRDSG)	The X coord. of all the input roads. Index into array for road N given by IRDSTRT(N).
IROADY (IRDSG)	The Y coord. of all the input roads. Index into array for road N given by IRDSTRT(N).
RDDMGE (IRDSG)	Fraction of road segment IRDSG damaged by air and ground fire

IROADX and IROADY are data base input variables that are defined in the road input deck. RDDMGE is a model generated variable that is used in subroutines DMGEVAL and OTHRDMG.

c) Term References:

Road segment is referenced first by determining which road the segment is a part of, then determining the endpoints of the segment relative to the start point of the road (see definition of road). For example, to examine the Kth segment of road N, the endpoints defined by {IROADX(IRDSTRT(N)+K-1),IROADY(IRDSTRT(N)+K-1)} and {IROADX(ORDSTRT(N)+K),IROADY(IRDSTRT(N)+K)} must be referenced. To examine the damage assessed against the Kth segment, reference the array element RDDMGE

(IRDSTRT(M)+K). Note that if IRDNPTS(N) gives the number of points comprising the entire road N, then the elements of arrays IROADX and IROADY, ranging from IRDSTRT(N) to IRDSTRT(N)+IRDNPTS(N)+1, would have to be referenced to completely define road N.

The following subroutines use road segment arrays:

ADJROM	RDSON
AIRHOTZN	ROADCHK
BRRDCHCK	ROADINP
DMBEVAL	SETIMP2
OTHRDMG	SPTALO
POINTGT	

3.122 ROAD TYPE

3.122.1 English-language Definition

a) Term Name:

Road Type

b) Term Description:

Road type is an attribute of the entity road. It distinguishes the kinds of roads being modeled by the CATTS math model currently three kinds of road are represented:

1) multi-lane all weather high speed road
2) single-lane all weather high speed road
3) single-lane all weather low speed road

The three road types will suffer different degrees of damage from air and ground ordnance. Road type one will experience relatively less damage over a given road segment than would road types two or three when damage is inflicted by the same kind of ordnance. Each road type has a characteristic width which is a factor in damage calculations.

c) Term References:

Road type is referenced by road number. Every road defined in the model has a code identifying which of the above three types it is. Road type information is used by the movement and firing functions of the math model. The former function is concerned with movement rates and the latter function is concerned with damage to roads.

3.122.2 Programmatical Definition

a) Term Name:

IRDTYPE(50)

b) Term Description:

IRDTYPE is a data base input variable that is defined in the road input deck. It is an integer array containing a code which distinguishes the kinds of road modeled by the CATTS math model. Currently the code may take on three values:

IRDTYPE(IRD) = type of road IRD ($1 \leq IRD \leq 50$)

1 multi-lane all weather high speed road

2 single-lane all weather high speed road

3 single-lane all weather low speed road

The code identifying road type can be used to reference the width of the road. Specifically the array IRDWDTH(IRDT) is an integer array containing the width in meters of the three types of road where IRDT is given by IRDTYPE(IRD) and ($1 \leq IRDT \leq 3$).

c) Term References:

Road type is indexed by road number. The array IRDTYPE is referenced by the following subroutines:

ADJROM
DMGEVAL
OTHRDMG
RDSON
ROADINP

3.123 ROUTE

3.123.1 English-language Definition

a) Term Name:

Route

b) Term Description:

A route is a series of connecting line segments generated by defining up to seven points in the area of operation. Routes may be pre-defined by input, or interactively defined by creating a special route or control measure route. A maximum of fifty routes may exist in the model simultaneously.

A unit instructed to follow a route will do so by moving directly from point to point in successive order. The unit must be traveling under movement code six. The route is followed to completion unless the unit encouters the enemy in an engagement, or is stopped by an obstacle.

c) Term References:

The data identifying a route is referenced by the route number (an integer between 1 and 50 inclusively). Routes are used by the movement function and the maneuver control function to direct units along prescribed courses of travel.

3.123.2 Programmatical Definition

a) Term Name:

IXPTH(7,50),IYPTH(7,50),NCDDP(50)

b) Term Description:

Routes are described by data entered into the integer arrays IXPTH, IYPTY, and NCDDP. There are three ways to enter data into these arrays:

1) pre-defined input
2) by maneuver control referencing a control measure route
3) by maneuver control having the controller choosing no more than seven points within the area of operation

A route consists of at most seven X-Y coordinate values stored successively in the arrays IXPTH and IYPTH. The a-ray NCDDP records the number of coordinate points comprising each route. For route IRTE ($1 \leq$ IRTE ≤ 50), and I($1 \leq$ I ≤ 7), the following definitions apply:

IXPTH (I,IRTE)	X-coordinate of the Ith point for the route IRTE
IYPTH (I,IRTE)	Y-coordinate of the Ith point for the route IRTE
NCDDP (IRTE)	Number of points describing the IRTEth route that units may be required to follow.

IXPTH, IYPTH, and NCDDP are data base input variables, interactively generated variables, as well as model generated variables. As data base input va iables, they are defined in the movement input deck; as interactively generated variables, they are created from the maneuver menu; and, as model generated variables, they are used in subroutine SPROUTE.

c) Term References:

The arrays defining routes is indexed by route number IRTE, where IRTE ranges from one to fifty inclusively. Routes are used by the following subroutines:

ARRIVE	NEWMOV
DIRMOV	OGDIR
MANEUVER	OG2UNI
MOVINP	SPROUTE

3.124 SAVE FILES

3.124.1 English-language Definition

a) Term Name:

Save Files

b) Term Description:

In the CATTS system, there are two save files; namely, the Replay file (DC,SAVE19) and the Restart file (DC,SAVE20). During each minute of the simulation, common blocks required for replay are dumped to save file DC,SAVE19 where the contents will be available for replay. Save file DC,SAVE19 has the capacity to hold 2 16,000-word blocks. At user-selectable 15-minute intervals, all other information is dumped to save file DC,SAVE20 where the contents will be available to restart or backup the simulation to that point. Save file DC,SAVE20 has the capacity to hold 3 16,000-word blocks.

c) Term References:

None

3.124.2 Programmatical Definition

(Not applicable)

3.125 SCENARIO

3.125.1 English-language Definition

a) Term Name:

Scenario

b) Term Description:

In the CATTS system, a scenario consists of:

1) A definition of both the Red and Blue forces, including unit type (i.e., mechanicah infantry, armor, mortar, artillery, air defense, combat engineering, etc.), numbers of personnel and equipments, as well as equipment performance.

2) A scheme of maneuver and fire support plan when appropriate.

Three scenarios have been implemented in CATTS. These are the FEBA GOLD, FEBA SILVER, and ATTACK scenarios. All three of these scenarios represent the Sinai region in the Middle East. FEBA GOLD represents the situation where the Red and Blue forces are deployed on opposite sides of the Suez Canal and the Red (enemy) firce attacks the Blue (friendly) force. FEBA SILVER represents a situation where the Blue (friendly) force, under attack by the Red (enemy) force, pulls back from the Suez Canal to the Mitla Pass region where it sets up an area defense. ATTACK represents a situation where the Blue (friendly) force counter-attacks the Red (enemy) force from its Mitla Pass positions and pushes the Red force back across the Suez Canal.

c) Term References:

None

3.125.2 Programmatical Definition

b) Term Description:

Four files define a scenario:

1) Card image scenario file; 2) Binary scenario file (filled by computer from card image file); 3) Prescheduled events file; and 4) Preplanned mission file (filled by computer from data in PPM deck on card image scenario file.

3.126 SECTION

3.126.1 English-language Definition

a) Term Name:

Section

b) Term Description:

A section is designated in the model by setting the size value for a specific unit to 2. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for section is "..".

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a size of 2 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 2 used to represent section.

3.126.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit deck for units 1-100, the operational grouping input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150. A section is designated for Unit I in the model by setting array ISIZE(I) to a value of 2 as part of the data base inputs. This value is never altered for units. A section is designated for operational group (I-100) by setting array ISIZE(I),($101 \leq I \leq 120$), equal to 2 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 2 for ISIZE(I) where I is in the range ($121 \leq I \leq 150$), represents the designation of section for red and blue adjacent units represented by those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.127 SECTOR

3.127.1 English-language Definition

a) Term Name:

Sector

b) Term Description:

Fire support sectors are defined as part of the scenario input deck. A fire-support unit is often required to restrict its fire to the direct support of a particular combat force or to targets located within a particular sector of the battlefield. In the CATTS automatic support fire logic, such a restriction is imposed by specifying the boundaries of a sector and requiring unit IU to limit its fire of weapon IEQ to targets lying within this sector. Sector boundaries are given as two lines parallel to the x-axis, as shown in Figure 3-5. Options are available (1) to fire only at targets located within the given sector or (2) to fire at close-support and interdictory-fire targets within the sector, but fire at other classes of targets wherever they may be found (see code Y in the definition of the fire support table (IFSTAB). The sector into which Unit IU is allowed to fire may be changed merely by changing the operational state of Unit IU.

c) Term References:

Fire support sectors are used as an option in the allocation of support fire weapons. The sectors are predefined in the input deck, and not changed.

3.127.2 Programmatical Definition

a) Term Name:

IYBDS(I,J)

b) Term Description:

IYBDS (I,J)	Lower (J=1) and upper (J=2) bounds on the sector in which weapon fire will be allocated where I is the index corresponding to IFSTAB I index, in terms of meters. Sectors are defined at initialization and not altered thereafter.

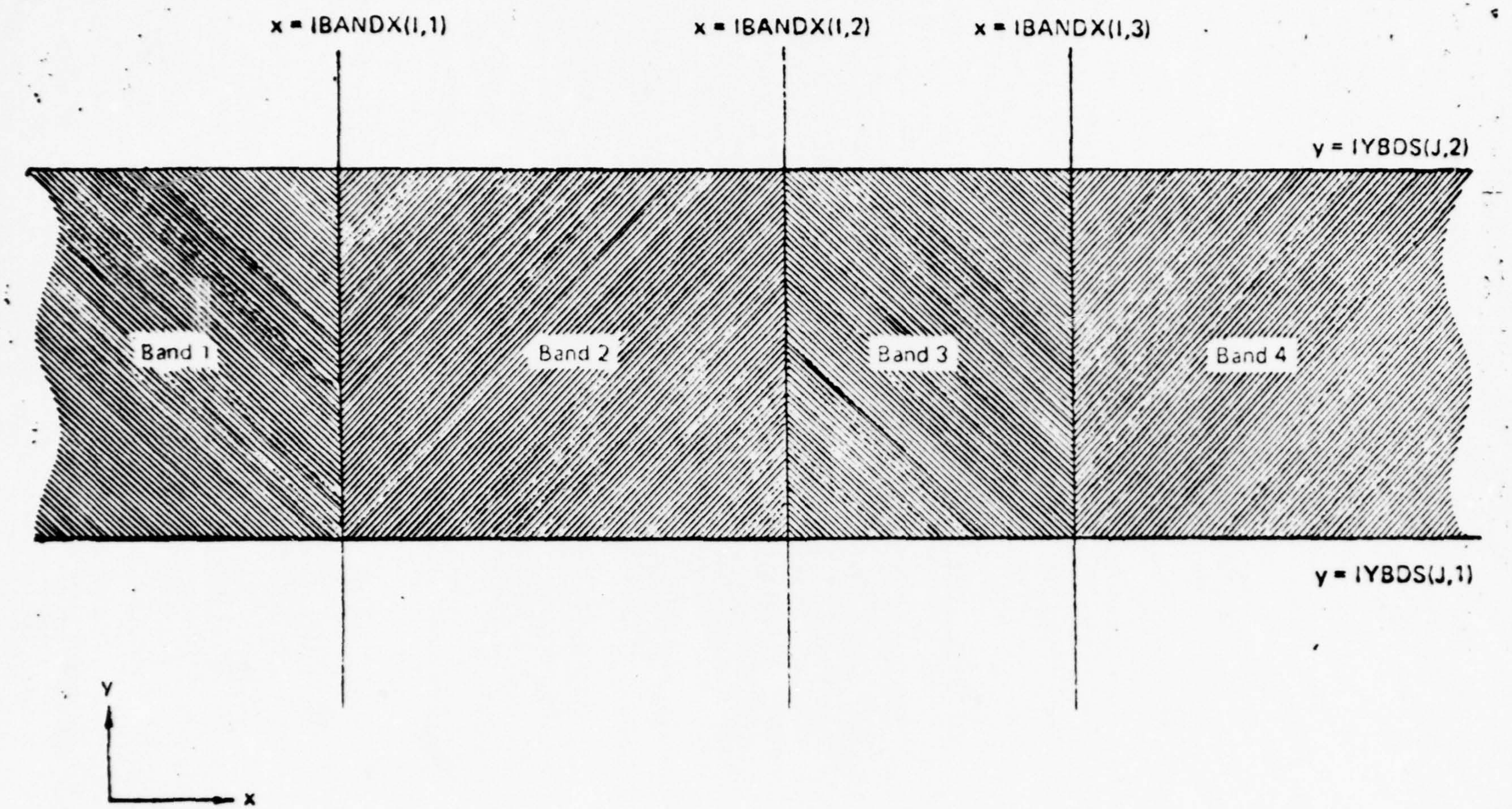


Figure 3-5. Fire-Allocation Sector and Fire-Support Target Bands

IYBDS(I,J) is a data base input variable that is defined in the fire support input deck.

c) Term References:

I corresponds to the entry in the IFSTAB table of code words. Therefore, I ranges between ($1 \leq I \leq 80$). The index J = 1 or 2. Three subroutines use IYBDS, one of which is to initially read in the sector data:

ENAREA
FSPINP
TGTCAT

3.128 SIMULATION CONTROL

3.128.1 English-language Definition

a) Term Name:

Simulation Control

b) Term Description:

The simulation control module of the command and control subsystem is used only by the main controller for controlling the CATTS game. Unlike other modules of the command and control subsystem, no event notice is created by simulation control. Instead, the actions requested by simulation control are taken in a more direct manner. For all control actions except freeze, the requested actions are stored in mailbox locations. Later, these actions are interpreted by the mathematical model via a CAL4 and appropriate steps are taken to comply with the various requests. For freeze, the simulation is halted by checkpointing the math model with program SIMCON. This foreground program then simply waits in a loop reading the control switch and releases when that switch is depressed.

Simulation control actions affect the operation of the command and control subsystem polling routine (subroutine POLL). Here, actions taken when the simulation control button is depressed must be coordinated with actions the mathematical model is taking. For example, when the mathematical model is restarted, the simulation control switch is depressed three times in the restart procedure. Since these simulation control actions are not requests for the simulation control menu, they must be ignored by the polling routine. In this way, the mathematical model, the polling routine, and the simulation control module are able to work in a coordinated manner.

c) Term References:

None

3.128.2 Programmatical Definition

(Not applicable)

3.129 SOIL

3.129.1 English-language Definition

a) Term Name:

Soil

b) Term Description:

Soil in the CATTS math model is represented by a table of data values which are used to affect movement of equipment (and thus units), and detection verdicts from unattended ground sensing devices. The table also contains data modeling the effect of luminous reflectance (by the soil) on visual detection. In short, soil is an entity defined by the following attributes:

1) rating cone index (wet and dry) affecting movement
2) detection degradation factors affecting unattended ground sensing devices
3) luminous reflectance factor affecting visual detection

Currently fifteen different types of soil can be modeled simultaneously. Each type is identified by an integer ranging from one to fifteen inclusively.

c) Term References:

The data describing soil is referenced by the soil type integer. This data is utilized by the movement and detection modules. Movement is concerned with trafficability of vehicular equipment. The detection logic deals with degradative effects of soil on vision and unattended ground sensors.

3.129.2 Programmatical Definition

a) Term Name:

SOIL(15,13)

b) Term Description:

SOIL is a data base input variable that is defined in the unattended ground sensor input deck. It is a floating point array containing data relating the effect of various soil types on equipment movement and detection. Column 1 of the array (SOIL(I,1) $1 \leq 0 \leq 15$) contains the rating cone indices of the various soil types when dry. The rating cone index is a positive number that may range from 0 to 170. Higher indices reflect better trafficability by vehicles. Column 2 of the array contains the rating cone indices when the soil type is wet. Columns 3 through 12 contain performance degradation factors for unattended ground sensors. These factors are fraction between zero and one inclusively. Column 13 contains fractions, also between zero and one inclusively, which model the factor of luminous reflectance. This factor is needed by the visual detection calculations. The following summaries the data required to model soil:

SOIL (ISOIL,J)	Characteristic data for soil type ISOIL:
	J=1 for RCI index when the soil is dry
	J=2 for RCI index when the soil is wet
	J=3 performance factor for the UGS type 1
	J=4 performance factor for the UGS type 2
	:
	:
	J=12 performance factor for the UGS type 10
	J=13 the reflectance of soil type ISOIL

c) Term References:

The array SOIL is referenced by soil type ISOIL, where ISOIL ranges from one to fifteen inclusively. For a given soil type, thirteen data values are given. The first two data values are rating cone indices, the next ten data values are performance degradation factors for unattended ground sensors, and the last data value is the luminous reflectance factor for the given soil type. The array SOIL is used by the following subroutines:

ADJROM
FORMAIN
LOSINP
SENSINP
VASCK
VISUAL

3.130 SPECIAL THREAT

3.130.1 English-language Definition

a) Term Name:

Special Threat

b) Term Description:

Any target unit in a special threat operational state is weighted more heavily as a potential target when an enemy unit prepares for fire allocation. This weighting factor is a function of unit type, color, and type of fire being allocated. The first and last special threat operational states are defined at initialization, and all operational state numbers between them and including them are considered special threat.

c) Term References:

The special threat factor is considered in the allocation of direct, indirect and support fire weapons against target units in a special-threat operational state.

3.130.2 Programmatical Definition

a) Term Name:

NSPTHR(I)

b) Term Description:

The special threat array, NSPTHR(I), is a data base input variable that is defined in the change of state input deck. It is defined as follows:

NSPTHR (I) — I = 1 is number of the first operational state in which a target is considered a special threat. I = 2 is number of the last such operational state. All operational state numbers in between are also considered special threat.

Operational states values are input at initialization and never changed. The current data base inputs zero into this array, making this capability ineffective.

c) Term References:

The index, I, to NSPTHR(I) is a dummy, where I = 1 or 2. Subroutine CBTVAL uses NSPTHR for direct and indirect, fire weapon allocation, and CLSPTG uses it for support fire allocation.

3.131 SQUAD

3.131.1 English-language Definition

a) Term Name:

Squad

b) Term Description:

A squad is designated in the model by setting the size value for a specific unit to 1. This causes the appropriate size display to appear above the tactical overview symbol for that unit when the tactical overview button is selected. The symbol for squad is ".".

c) Term References:

The foreground graphic display programs refer to the size of each unit, unit number being the direct reference.

Those units designated a size of 1 have the symbol used as part of the tactical overview display. Operational groups and adjacent units also have a size associated with them, with the value 1 used to represent squad.

3.131.2 Programmatical Definition

a) Term Name:

ISIZE(I)

b) Term Description:

ISIZE(I) is a data base input variable that is defined in the unit input deck for units 1-100, the operational grouping input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150.

A squad is designated for Unit I in the model by setting array ISIZE(I) to a value of 1 as part of the data base inputs. This value is never altered for units. A squad is designated for operational group (I-100) by setting array ISIZE(I),(101 $\leq$ I $\leq$ 120), equal to 1 to designate the appropriate size of the op group. This designation is made via data base inputs for pre-defined op groups, or as a result of interactively defining new op groups through the task organization menu. A value of 1 for ISIZE(I), where I is in the range (121 $\leq$ I $\leq$ 150), represents the designation of squad for red and blue adjacent units represented by those unit numbers.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.132 STATUS REPORT

3.132.1 English-language Definition

a) Term Name:

Status Report

b) Term Description:

There are three status reports associated with the CATTS mathematical model; namely, the special status report, and the Army and TRW versions of the mathematical model status report. The special status report is available upon request, at a controller's Superbee, for any single unit. This report gives the unit's location, speed, elevation (or altitude), current personnel levels, current equipment and ammunition levels, and the current fuel level. The Army version of the mathematical model status report is a real-time status report available at 15-minute intervals. This report outputs the current personnel, equipment, ammunition, and fuel levels for a unit plus the change in these levels since the preceding 15-minute report. The TRW version of the mathematical model status report is available at user-selected intervals and provides the same type of information contained in the Army version plus comprehensive summary data such as firing summaries, detection summaries, etc.

c) Term References:

None

3.132.2 Programmatical Definition

(Not applicable)

3.133 SUPPRESSION CRITERIA

3.133.1 English-language Definition

a) Term Name:

Suppression Criteria

b) Term Description:

Each unit, depending on what type it is, its operational state, and whether it is Red or Blue, will have its level of suppression: (1) determined by a function of any one of four variables, (2) set to zero, or (3) left alone. This is done by matching the unit to an entry in the Suppression Criterion Selection Table described fully in the next section. This entry then designates the suppression criterion to use, i.e., the independent variable used to determine suppression, and the number of the input (empirical) suppression curve to use with this criterion. If the criterion is 0, the suppression factor is set to zero; if there is no matching entry for the unit in the Suppression Criterion Selection Table, the unit's suppression factor remains unchanged. The four criteria are imbedded in the program logic of the suppression function. These criteria are:

1) Density of incoming support fire ordnance per unit time (number of support fire rounds per unit time per square meter).
2) Rate of loss of personnel in unit if unit were unsuppressed. (Men lost/unit time).
3) Fraction of a unit's initial strength lost per unit time if unit were unsuppressed.
4) Fractional reduction in unit's initial strength.

c) Term References:

Suppression criteria are programmed equations for determining the function of a unit suppressed. The table identifying which criterion is to be used is discussed at length in the next section. It is initialized at system start-up and not changed. The table uses a unit's type, operational state, and color to determine what criterion and which suppression function is to be used.

3.133.2 Programmatical Definition

a) Term Name:

ISSLCT(I)

b) Term Description:

ISSLCT(I) is a data base input variable that is defined in the suppression input deck. The four criteria listed above in the English-language Definition are computed with the following program variables:

Criterion 1: SWFU(IU)/Unit IU Area

Criterion 2: $\frac{DPERSU(IU)}{DTIME}$

Criterion 3: $\frac{DPERSU(IU)/PERSI(IU)}{DTIME}$

Criterion 4: $\frac{PERSI(IU) - PERS(IU)}{PERSI(IU)}$

where

SWFU(IU) = number of support fire rounds falling on Unit IU during last time interval

DTIME = time step increment

Unit IU Area = area occupied by Unit IU (sq. meters)

DPERSU(IU) = number of men which would have been lost in unit IU during last time step if unit IU were unsuppressed

PERSI(IU) = initial number of men in unit IU

PERS(IU) = number of men in unit IU at time t

SWFU(IU) and DPERSI(IU) are computed by the firing routines and provided as input to the suppression routine. These equations are hard-coded in subroutine SUPRES. This subroutine determines, from the Suppression Criterion Selection Table, which of the above four criteria is to be used, if any.

ISSLCT (I) Ith entry in the suppression criterion selection table (1 < = I < = 100) of the form CXYYZSS, where
C=Number of the suppression criterion
X=Unit type
YY=Unit operation state
Z=1 or 2 (Red or Blue)
SS=Number of the suppression curve to use.

If no match is found in the table, the unit's suppression remains unchanged. If a criterion of zero is found, suppression is set to zero. Otherwise, the indicated criterion is computed and used as the abscissa of the designated suppression curve. The corresponding ordinate of this curve is used as the unit's suppression.

c) Term References:

Table ISSLCT(I), the suppression selection table, has a dummy index, I($1 \leq I \leq 100$).

Subroutine SUPRES contains the equations for all four criteria mentioned above, performs the table search to determine which criterion to use, and computes the unit's suppression from the designated function.

3.134 SUPPRESSION EFFECTS

3.134.1 English-language Definition

a) Term Name:

Suppression Effects

b) Term Description:

The suppression factor for a unit is a result computed from the suppression function (see the term "suppression criteria"). This factor is a number between 0 and 1 indicating the fraction of men temporarily not performing in their prescribed manner. The model implements this effect in the ground fire module by moving the suppressed fraction of men and equipment from each of modes 2 through 8 into mode 1 (see the term "modes"). Elements in mode 1 normally will not be combat-effective regarding their ability to fire and move, but will also not be very vulnerable to the enemy's weapons. Factors such as capability to fire and move, and vulnerability to enemy fire, are defined at initialization time for each of the 8 modes, including mode ˉ

c) Term References:

Suppression factor of a unit is a unit attribute resulting from the suppression module processing. The effects of the suppression factor on a unit's combat effectiveness is determined in the subsequent minute in the ground fire module.

3.134.2 Programmatical Definition

a) Term Name:

SIGU(IU)

b) Term Description:

SIGU(IU) is both a data base input variable and a model generated variable. As a data base input variable, it is defined in the unit input deck, and, as a model generated variable, it is used in subroutine SUPRES.

The result of subroutine SUPRES is placed in array SIGU(IU) for each unit, IU, every time step. SIGU(IU) is a floating point number in the range [0,1], defined as:

SIGU (IU) Suppression factor: Fraction of men and equipment in unit IU who are suppressed.

c) Term References:

SIGU(IU) is indexed by unit, where ($1 \leq IU \leq 100$). Subroutines SUPRES sets this array for all units, and subroutines AIRCAS, ORGFIR, and SPTALO primarily use it to determine fire and movement capability for air defense, direct and indirect fire, and support fire weapons, respectively

3.135 TACTICAL OVERVIEW

3.135.1 English-language Definition

a) Term Name:

Tactical Overview

b) Term Description:

Tactical overview is a graphic display console selection which provides for the display of standard military symbols depicting the unit arm/branch/duty and size of each unit in the data base. The designations of unit arm/branch/duty and size are data base inputs. Size, therefore, is not necessarily a function of the number of personnel in a unit, or the area covered by the unit. When the tactical overview button is selected, all units of the color desired for display will appear with their appropriate symbol.

c) Term References:

The reference to the variables describing unit arm/branch/duty and size, which together determine the exact symbol to be displayed, is the unit number. The array containing size data includes additional size information for operational groups and adjacent units which is not applicable to the tactical overview display.

3.135.2 Programmatical Definition

a) Term Name:

ITPPL(I) and ISIZE(I)

b) Term Description:

ITPPL(I) is a data base input variable that is defined the the unit input deck. ISIZE(I) is both a data base input variable and an interactively generated variable. As a data base input variable, ISIZE is defined in the unit input deck for units 1-100, the operational grouping input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150. As an interactively generated variable, ISIZE is created from the tasking organization menu for units 101-120.

The arrays ITPPL and ISIZE together provide data to the foreground graphic display programs to determine the type of symbol to be drawn for Unit I as part of the tactical overview display. The values for arrays ITPPL and ISIZE are input as part of the data base for all units, and the symbols to be drawn for each value are listed in Table 3-4.

c) Term References:

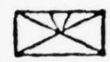
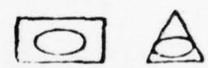
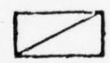
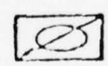
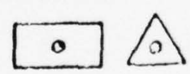
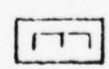
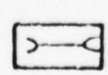
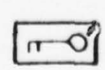
ITPPL(I) is referenced solely by a unit number index and is used in subroutines UNINP and FORRAM. The range of I is $(1 \leq I \leq 100)$.

ISIZE(I) is referenced primarily by a unit number index; hence it is primarily an attribute of a unit. The range of I is $(1 \leq I \leq 150)$, where the first 100 values of ISIZE correspond to units. For I such that $(101 \leq I \leq 120)$, values of ISIZE correspond to the size of operational groupings. For I such that $(121 \leq I \leq 150)$, values of ISIZE correspond to the size of adjacent units, 15 allocated for red and 15 for blue.

The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

Table 3-4. Tactical Symbols

UNIT ARM/BRANCH/DUTY	SYMBOL	ITPPL
Infantry		1
Infantry, Mech		2
Infantry, Airmobile		3
Infantry, Airborne		4
Armor		5
Cavalry		6
Armored Cavalry		7
Anti-Tank		8
Artillery, Towed		9
Artillery, SP		10
Air Defense		11
Engineer		12
Signal		13
Maintenance		14
Medical		15
Ordnance		16
Quartermaster		17

(-) Preceding ITPPL causes an OP symbol (△) to be drawn.

Table 3-4. Tactical Symbols (Continued)

UNIT SIZE	SYMBOL	ISIZE
Squad	•	1
Section	••	2
Platoon	•••	3
Company/Battery/Troop	1	4
Battalion	11	5
Regiment	111	6
Brigade	X	7
Division	XX	8
Corps	XXX	9
Army	XXXX	10
Army Group	XXXXX	11

(-) Preceding ISIZE causes a staff to be drawn on symbol, meaning unit is a CP.

3.136 TARGET

3.136.1 English-language Definition

a) Term Name:

Target

b) Term Description:

A target in the CATTS model is any entity that can be fired at by ground or air fire. A target can be one of the following several types:

- opposing unit
- X,Y point
- road segment
- bridge
- aircraft

Fire directed via command and control can be of three types: direct (or indirect ground fire; ground support fire; air-delivered fire. Direct (or indirect) ground fire can be commanded only at opposing units. Ground support and air-delivered fire can be directed against any one of the four types. Automatic ground fire (direct, indirect, and support) against ground targets can only be allocated and fired at opposing units. Air defense fire can only be allocated against opposing aircraft. It is possible to damage one's own ground units by firing at X,Y point targets by means of ground support or air fire commands, if the friendly units are in the near vicinity of the ordnance impact. It is also possible to accidentally damage one's own aircraft if that aircraft flies through an area of support fire. These latter two cases of accidental damage to one's own units are examples of non-target units incurring casualties.

c) Term References:

Target is basically a unit number specification, but for command and control, it can be used also as a bridge, road, or X,Y point indicator and actual bridge or road number.

3.136.2 Programmatical Definition

a) Term Name:

II

b) Term Description:

Variable II is both an interactively generated variable and a model generated variable. As an interactively generated variable, it is created from the air and fire menus, and, as a model generated variable, it is used in subroutines BDTGTS, CBTVAL, FIRALO, GENFIR, NOTGT, ORDPRI, SETINP1, SPTALO, WTSETS, and WTSUB. It is used throughout the ground fire, air defense, and air-delivered ordnance routines FIRALO and SPTALO, AIRCAS, and ADW, respectively, to represent the target unit or designation number. This variable takes on the following values for targets of ground weapons and air defense weapons:

II	Interpretation
0	X,Y point
1-99	Unit number (1-99)
101-116	Bridge number (1-16)
201-700	Road segment number (1-500)

For air-delivered ordnance, the values of II are:

II	Interpretation
1-99	Hard target (unit number, 1-99)
201	Area target (X,Y point)
202	Road segment
203	Bridge

The non-unit targets above are indicators, and the air-delivered ordnance subroutines OTHRDMG, DIDITHIT, and POINTGT determine which specific entity is affected.

c) Term References:

II is an index to unit, bridge, and road segment arrays, and is also used as an indicator in the Air Module. When used as a bridge

index, 100 is subtracted from II to directly access the bridge arrays; as a road index, 200 is subtracted from II to directly access the road segment arrays. Subroutines

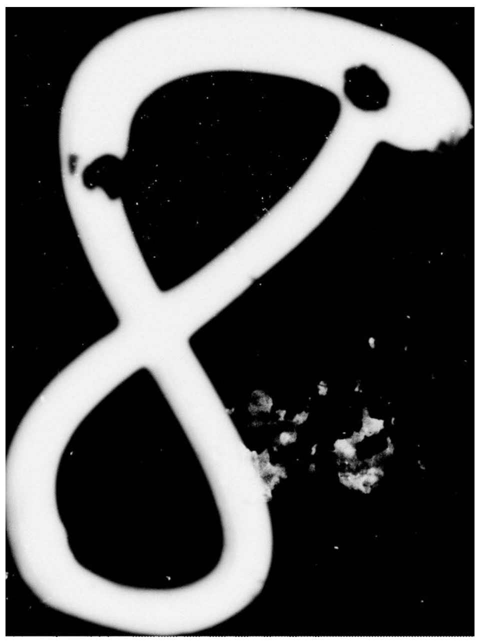
FIRALO
SPTALO
AIRCAS
ADW

are the primary users of II as a target index.

AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

6 OF 8
AD A038796



3.137 TARGET ELEMENTS

3.137.1 English-language Definition

a) Term Name:

Target Elements

b) Term Description:

Target elements are used in CATTS to refer to the opposition's personnel and/or equipment types that a weapon desires to fire at. These preferences are specified by up to three choices of primary targets, and three choices for secondary targets. If personnel is specified as a primary target, no equipment can be specified for primary target, although equipments can be specified as secondary target elements; likewise for personnel specified as secondary target. Only personnel and/or equipments designated as primary or secondary will be considered to determine the allocation of fire of a given weapon type. However, the fire will be effective against all personnel and equipment in the unit being fired on for which the weapon's effects are defined. The target element priorities merely help determine fire allocation. Separate primary and secondary target elements are accumulated so that a weapon can be allocated against only primary elements, and, if none are available, secondary elements. If two or three equipments are specified as primary or secondary target elements, the two or three are treated equally within their category.

c) Term References:

Both primary and secondary target elements are equipment attributes used primarily by the Ground Fire Module to allocate fire against targets, and used also by the Air Module to determine the best ordnance for an air strike.

3.137.2 Programmatical Definition

a) Term Name:

NPTE(IEQ,J)
NSTE(IEQ,J)

b) Term Description:

Primary and secondary target elements are represented in arrays that are data base input variables defined in the equipment input deck. These arrays are:

NPTE(IEQ,J) Equipment numbers of two primary target types for weapon IEQ; -1 implies a personnel target; for air ordnance

IEQCOD(IEQ)= -2), NPTE(IEQ,J) represents a priority number for selecting air ordnance against a designated target type, where

J=1 ⇒ non-armored unit type
J=2 ⇒ lightly armored unit type
J=3 ⇒ heavily armored unit type

An entry in either NPTE or NSTE must be one of the following:

ENTRY	INTERPRETATION
-1	personnel
0	no more entries
valid eqpt. #	specified equipment # is a target element

NSTE(IEQ,J) Equipment number of two secondary target types for weapon IEQ; -1 implies a personnel target;

For air ordnance (IEQCOD(IEQ)= -2), NSTE(IEQ,J) represents a priority for selecting air ordnance against a designated target type, where

J = 1 ⇒ area target
J = 2 ⇒ road
J = 3 ⇒ bridge

c) Term References:

NPTE(IEQ,J) and NSTE(IEQ,J) are indexed primarily by equipment, where ($1 \leq IEQ \leq 80$), secondly by a dummy index, J, where ($1 \leq J \leq 3$). These arrays are primarily used by the following subroutines:

CBTVAL
SPTALO
TGTLST
ORDPRI

3.138 TARGET ELIGIBILITY

3.138.1 English-language Definition

a) Term Name:

Target Eligibility

b) Term Description:

For direct and indirect fire weapons, an indicator is set for each weapon unit/target unit pair examined to indicate whether or not the target unit is a valid target for the weapon unit. This determination is the principal one in forming a weapon/target set.

c) Term References:

Target eligibility is an indicator used only in the direct (and indirect) fire allocation processing of the Ground Fire Module.

3.138.2 Programmatical Definition

a) Term Name:

KELG

b) Term Description:

KELG is a model generated variable that is used in subroutine FIRELG. It is defined as:

1 $\Rightarrow$ target unit is eligible for fire by this weapon unit
2 $\Rightarrow$ target unit is not eligible for fire by this weapon unit

For a non-engaged firing unit, KELG is set to 2 when a direct fire weapon (KFICTR=1) is considered and a friendly unit blocks the line of sight. For an engaged firing unit, KELG is set to 2 when:

(1) the status code of the target unit is zero, and the distance to the local enemy FEBA for the target unit is greater than MAXID for the firing unit.

(2) The status code of the target unit is one, and the status code of the firing unit is zero.

Under all other circumstances, KELG = 1.

c) Term References:

Subroutine FIRELG sets the indicator KELG for use by subroutine WTSETS in the making of weapon-target sets.

3.139 TARGET LIST

3.139.1 English-language Definition

a) Term Name:

Target List

b) Term Description:

In the formation of weapon-target subsets for the automatic allocation of direct and indirect fire weapons, prioritized target lists are formed for each unit in the weapon subset. This list contains a maximum of 8 targets for a given weapon, and represents the target units that the specific weapon unit will fire at with the weapon under consideration. If more than eight targets are eligible for fire, the units with the closest range to the firing unit are retained.

For commanded fire, a separate target list is retained as part of the fire control table (see discussion of fire control table in next section). This table also holds a maximum of eight targets, but also includes the number of rounds or percentage of fire commanded, and the duration of the fire. The fire control table target list is processed at the beginning of both direct (and indirect) fire processing and support fire processing.

c) Term References:

Target lists are attributes of specific weapons in specific units. They are calculated each time step before fire is allocated.

3.139.2 Programmatical Definition

a) Term Name:

ITET(IEQ,J),ITGTLST(I,J)

b) Term Description:

The two types of target list arrays are defined as follows:

ITET (I,J)	Target eligibility table, for each entry in weapon-unit set I, a list containing all Jth eligible target units

ITGTLST (I,J) Target list for the Ith entry on the IFIROVRD table for up to 8 targets (J=1 to 8) packed as follows:

Bits 0- 9 -- target number
0 XY point
1-100 unit number
101-200 bridge number+100
201-700 road segment number+200

Bits 10-17 -- duration in minutes (used by RPM only)

Bits 18-31 -- number of rounds to fire, or percent of fire * 100

Entries 9 and 10 ((I,9) and (I,10)) are 8 byte-size pointers to the impacting fires array (IXYIM), one for each of the 8 targets. Word 9, byte 0 corresponds to target 1, etc.

ITET is a model generated variable that is used in subroutine TGTLST, and ITGTLST is an interactively generated variable that is created from the fire menu.

c) Term References:

Subroutine TGTLST sets ITET (I,J), which has the index from the weapon set entry I(IW(I)), ($1 \leq I \leq 30$). Since up to eight targets can be listed for a weapon unit, ($1 \leq J \leq 8$). Subroutines WTSETS, WTSUB, and FIRALO are principal users. Array ITGTLST(I,J) part of the fire control table, which has 100 entries. Each entry can have up to 8 different targets, and two other words are used as pointers to the impacting fire table. Therefore, ITGTLST is a 100 x 10 array, where ($1 \leq J \leq 8$) represents the eight possible targets. Subroutine FIRSORT sets ITGTLST. Subroutines FIRALO and SPTALO are principal users.

3.140 TARGET WEIGHTING FACTORS

3.140.1 English-language Definition

a) Term Name:

Target Weighting Factors

b) Term Description:

Weighting factors are used in the pre-calculation of a target unit's value to a specific opposing force weapon being allocated. The opposing force weapon has primary and secondary target elements (see discussion of this term) defined for it. If a target unit has one or more of these elements, its value as a target is computed. If the weapon has personnel designated as the target element, the weighting factors for each vulnerability class are used to compute a weighted value for personnel. The number of personnel in each class is multiplied by the weighting factor, and accumulated over all vulnerability classes. If one or more equipment types are designated as the target elements, the weighted value for equipment is computed by multiplying the weighting factor times the number of pieces of each equipment type. Separate accumulations are made for primary and secondary target elements.

c) Term References:

Equipment weighting factor is an equipment attribute; personnel vulnerability class weighting factor is an attribute of personnel vulnerability class. Both are used exclusively in the Ground Fire Module to allocate both direct (and indirect) and support fire weapons.

3.140.2 Programmatical Definition

a) Term Name:

UBE(IEQ)
WPVC(IPVC)

b) Term Description:

Personnel and equipment weighting factors are defined as follows:

UBE(IEQ)	Weighting factor: importance of equipment type IEQ as target
WPVC(IPVC)	Weight of a man as a target element for each personnel vulnerability class IPVC.

These weighting factors are data base input variables that are input at system initialization and not changed. Both arrays, although floating point numbers, are integer values as opposed to fractions. UBE is defined in the equipment input deck, and WPVC is defined in the first deck (miscellaneous variable deck).

c) Term References:

Array UBE(IEQ) is indexed by equipment, where ($1 \leq IEQ \leq 80$). Array WPVC(IPVC) is indexed by personnel vulnerability class number, where ($1 \leq IPVC \leq 6$), although only ($1 \leq IPVC \leq 4$) is defined. Subroutine EQIPNR is called by primarily CBTVAL and SPTALO to compute equipment target unit weights. Subroutine STEP pre-calculates the personnel weights of all units at the end of each time step for the succeeding time step.

3.141 TASK ORGANIZATION

3.141.1 English-language Definition

a) Term Name:

Task Organization

b) Term Description:

Task organization pertains to the interactive capability of gathering individual units to form a new operational group. The operational group is created to allow groups of units to be maneuvered as a single entity. This permits units to group in formation in order to move or deploy against enemy forces. Maneuvering groups of units as a single body relieves the controller of having to issue commands to each of several units to accomplish an objective. Also, task organization realistically models the behavior of units which are part of a coordinated team, and therefore would not normally behave independently of each other.

c) Term References:

Task organization is referenced when the controller brings up the task organization menu. The menu allows the controller to:

1. create a new operational group
2. delete a unit from an operational group

If an operational group is created via the menu before model initialization is complete, the units chosen for the group will be immediately relocated to the desired locations relative to one another. If the operational group is created via the menu after model initialization, the units in the group will move toward their designated locations with normal rates of movement.

3.141.2 Programmatical Definition

a) Term Name: INOG(IU)

b) Term Description: INOG(IU) operational grouping (1-30) to which ground unit IU belongs (=0 means no op group). It is set on the data base and changed only via interactive on table driven C & C.

c) Term References: Used for gound units in the following subroutines:

ADJDIR, ARRIVE, CHGOPN, DEPLOC, DEPLOY, DIRMOV, ENCTR, FIXOGCDS, FWDLIN, JOINOG, LEAVEOG, MANEUVER, MOVE, MOVMNT, NEWENG, NEWFWDUN, NEWMOV, NGARGN, OGCENTER, OGHFRONT, OGLOC, OGLOC2, OGTYPE, OG2UNI, OPPLAN, OVERUN, REL2FWDU, RMVOPGP, SETOGCDS, STATREP, STATREP1, STEP, STLKUP, TASKORG, TGTCAT, UNINP, and WTHDRW.

3.142 TERMINATION OF GAME

3.142.1 English-language Definition

a) Term Name:

Termination of Game

b) Term Description:

A CATTS simulation run is normally terminated in two ways:

1) When the principal instructor uses the simulation control menu to terminate the game.

2) When the input maximum simulation time is reached.

When the game is terminated normally, a post game summary consisting of a final status report and a casualty report will be automatically printed on the line printer. It is the design of the Simulation Control Submodule that when there are no Replay and Restart files, the simulation will not be terminated and will run. However, the restart and replay capabilities will be deactivated for that run and a RAM alert sent to the controller stating that the restart/ replay capability is not available.

c) Term References:

None

3.142.2 Programmatical Definition

(Not applicable)

3.143 TERRAIN

3.143.1 English-language Definition

a) Term Name:

Terrain

b) Term Description:

Terrain representation in the CATTS math model considers four basic factors. These factors are explicitly defined by data inputs (i.e., they must be specified by the user). The factors are as follows:

1) Relief
2) Vegetation
3) Soil
4) Obstacles

These factors are defined elsewhere in this document.

c) Term References:

Terrain representation is used mainly for line of sight calculations, and determination of movement rates for ground units. Terrain data is also used extensively by the target acquisition module, which computes detection verdicts. Furthermore, terrain provides information to deal (implicitly) with the military aspects of cover and concealment, fields of fire, avenues of approach, etc.

3.143.2 Programmatical Definition

(Not applicable)

3.144 TIME STEP

3.144.1 English-language Definition

a) Term Name:

Time Step

b) Term Description:

A time step is defined in CATTS to mean the unit of discrete, equal time increments in which a series of events are processed in a given fixed sequence as though they were performed concurrently. These time steps are made small enough so that it may be reasonably assumed that significant changes in the way units operate will not occur during one time step. Certain exceptions to this rule are permitted, particularly with regard to the arrival of units at prescribed destinations and their dispatch to new destinations. In general, however, the nature of operations during a time step is considered to be uniform. Such factors as personnel, equipment, and ammunition levels; locations; rates and directions of movement; allocation of fire; and operational and movement codes are updated at the end of each time step. Air movement and interaction with ground forces occur much more rapidly. Air units are, therefore, updated four times in each time step, and are graphically displayed over four segments each time step.

A time step for the CATTS model is equal to one minute of simulated time. The model can be made to execute one minute of simulated time in one minute of actual clock time. Air unit positions and interactions are updated every fifteen seconds.

c) Term References:

None

3.144.2 Programmatical Definition

(Not applicable)

3.145 TRAVEL CODE

3.145.1 English-language Definition

a) Term Name:

Travel Code

b) Term Description:

The travel code is an attribute of the entity unit which indicates the following:

1) if the unit is a ground unit, it indicates the unit's intentions with respect to the enemy
2) if the unit is an air unit, it indicates whether the air unit is active

The intentions of a ground unit include:

1) the unit is trying to avoid an engagement with the enemy
2) the unit is neither avoiding nor seeking the enemy, but will engage the enemy if necessary
3) the unit is seeking an engagement with the enemy

For air units only, the travel code and the status code determine whether the air unit is active (i.e., conducting a mission). In particular, all air units have a travel code of four, but active air units have a status code of minus one, and inactive air units have a status code of zero.

c) Term References:

Travel code is an attribute of unit, thus it is referenced by unit number. The travel code is used principally by the movement, engagement, firing, and air modules.

3.145.2 Programmatical Definition

a) Term Name:

ITRAV(100)

b) Term Description:

ITRAV is a data base input variable, an interactively generated variable, as well as a model generated variable. As a data base input variable, it is defined in the unit input deck; as an interactively generated variable it is used in subroutines NEWMOV, OBSUPDAT, and OG2UNI. ITRAV is an integer array which indicates for each unit the unit's intentions with respect to the enemy.

ITRAV(IU) = travel code for Unit IU ($1 \leq IU \leq 100$);

1 - unit avoids engagement if possible

2 - unit neither desires nor avoids engagement until destination is reached

3 - unit seeks engagement

4 - air unit (used in conjunction with ISTATU(IU))

c) Term References:

ITRAV is indexed by unit number IU, where IU ranges from one to one-hundred inclusively. ITRAV is used in the following subroutines:

AIREVENT	MOVE
AIRHOTZN	NEWMOV
AIRMOV	OBSUPDAT
AIRMOV2	OGDIR
CK4XING	OG2UNI
CLSPTG	OVERUN
DIRMOV	SPTFIR
ENCTR	STATREP
FIRELG	STATREP1
FORMAIN	UNINP
FORRAM	UNI2OG
MANEUVER	

3.146 UNATTENDED GROUND SENSOR (UGS)

3.146.1 English-language Definition

a) Term Name:

Unattended Ground Sensor (UGS)

b) Term Description:

Unattended ground sensors are defined in CATTS by the following attributes:

1) type of sensor
2) radius of the sensor's field (meters)
3) X,Y coordinates of the sensor's location (meters)

Additional attributes are supplied by the UGS type, which is defined as a separate term.

The processing of unattended ground sensors results in the generation of Superbee alerts for all opposing units detected by the sensors. UGS detections are not a factor in unit-unit detection.

c) Term References:

Unattended ground sensors are entities in the model consisting of the attributes listed above. They are processed in the detection module each time step. Up to 10 UGS are allowed in the model. Eight are defined in the referenced data base.

3.146.2 Programmatical Definition

a) Term Name:

An unattended ground sensor is a model entity represented by the following program variables:

IUASFT(IUGS,ICOLOR)
UASFRS(IUGS,ICOLOR)
UASFX(IUGS,ICOLOR)
UASFY(IUGS,ICOLOR)

b) Term Description:

A programmatic description of an unattended ground sensor consists of the following program variables:

IUASFT (IUGS,ICOLOR)

Sensor field type for IUGS TH UGS field for each force
ICOLOR=1 red force
ICOLOR=2 blue force

UASFRS (IUGS,ICOLOR)

Radius for the IUGS TH UGS field in meters
ICOLOR=1 red force
ICOLOR=2 blue force

UASFY (IUGS,ICOLOR)

Y-Coordinate for IUGS TH UGS field in meters
ICOLOR=1 red force
ICOLOR=2 blue force

These variables are data base input variables that are defined in the UGS input deck. The variables UASNAME,UASTRS, and SOIL are additional UGS attributes grouped under the term "unattended ground sensor types".

c) Term References:

All four variables listed above are indexed primarily on unattended ground sensor number, IUGS, where ($1 \leq IUGS \leq 10$), secondly on color, ICOLOR, where ($1 \leq ICOLOR \leq 2$).

Subroutine UASCK processes all UGS in the model. Any opposing enemy unit found within range of an UGS, after sensor performance is degraded due to soil type, is flagged. Subroutine DALERT generates Superbee alerts for all such units.

3.147 UNATTENDED GROUND SENSOR (UGS) TYPE

3.147.1 English-language Definition

a) Term Name:

Unattended Ground Sensor (UGS) Type

b) Term Description:

Seven different unattended ground sensor types are currently modeled in CATTS (a maximum of ten is allowed), consisting of the following:

- disturbance
- seismic-type 1
- seismic-disturbance
- seismic-type 2
- electro-magnetic
- passive IR
- acoustic-seismic

Each type is characterized by specific data input at model initialization time. This data consists of:

- UGS type name
- UGS type maximum detection ranges for noisy and quiet units
- performance degradation factor for the UGS type as a function of soil condition.

If a unit is within the sensor field radius (see UGS discussion), it is detected. If a unit is outside the maximum detection range of the unit type, it is not detected. If a unit is between these two ranges, the maximum detection range is reduced by the soil degradation factor. If the unit is within this degraded range, it is flagged for a Superbee alert.

c) Term References:

The attributes of UGS type are also attributes of UGS. The ground detection module performs all processing for UGS.

3.147.2 Programmatical Definition

a) Term Name:

UNASNAME(I,IUGST)
UASTRS(I,IUGST)
SOIL(ISOIL,IUGST)

b) Term Description:

The following UGS types are modeled in CATTS and referenced when index IUGST is set to the associated numerical value:

1 = disturbance
2 = seismic - type 1
3 = seismic-disturbances
4 = seismic-type 2
5 = electro-magnetic
6 = passive IR
7 = acoustic-seismic

The above number is used as an index into the following arrays, which are data base input variables defined in the UGS input deck, and which together define UGS type:

UASNAME (I,IUGST)	The Ith word for the name of the UGS type IUGST
UASTRS (I,IUGST)	The range for the UGS type I
	I = 1 If the unit being detected will not produce noise than a truck.
	I = 2 Otherwise
SOIL (I,J)	Characteristic data for soil type I:
	J=1 for RCI index when the soil is dry
	J=2 For RCI index when the soil is wet
	J=3 Performance factor for the UGS type 1
	J=4 Performance factor for the UGS type 2
	:
	:
	J=12 Performance factor for the UGS type 10
	J=13 The reflectance of soil type I

The soil performance factors are used to degrade the maximum detection range of an UGS type as a function of the soil type of the unit being examined.

c) Term References:

Variables UASNAME and UASTRS are primarily indexed on unit type, which is the second of the two indices, where ($1 \leq IUGS \leq 10$). The first index for UASNAME is a dummy, I ranging from ($1 \leq I \leq 3$), to accommodate a 12 character name (4 characters per 32-bit word). The first index for UASTRS is a dummy, I, where ($1 \leq I \leq 2$), and is defined in the above description. The SOIL array is indexed primarily on soil type, secondly on (UGS type + 2), so that values for the second index ranging between 3 and 12, inclusive are for UGS type 1 through 10, respectively, for a given soil type.

Subroutine UASCK performs all UGS computations, and subroutine DALERT issues Superbee alerts resulting from UGS detections.

3.148 UNIT

3.148.1 English-language Definition

a) Term Name:

Unit

b) Term Description:

A unit is any collection of personnel and equipment physically located together and operating to accomplish an objective. Units are the basic functioning elements with respect to movement, detection, engagement, and firing. They may operate independently, or gather together in a group to accomplish a task in a coordinated effort. Presently, a maximum of one-hundred units may exist simultaneously in the model.

Though units share common attributes, there are no prescribed standard size and characteristic for units. They may be large or small, may have different types and amounts of equipment, and may contain different numbers of personnel. Units may be classified according to general type, but units of the same type need not be the same nor have the same organization, equipment, or personnel. A maximum of twenty different types of units may be specified.

c) Term References:

A unit is referenced by accessing the information defining the attributes of the unit. Each attribute can be examined by specifying the unit number. This number is an identification integer ranging in value from 1 to 100 inclusively. The unit is the most fundamental entity in the CATTS math model. Because of this, it is described in much greater detail, requiring an extensive set of attributes. References to unit attributes, or the concept of unit are made implicitly and explicitly by all major modules of the CATTS math model.

3.148.2 <u>Programmatical Definition</u>

a) Term Name:

BDIR(100,2)	IADWUNIT(4)	IOBALRT(4)
BLAVGAS(100)	IAGDET(4,10,6)	IOBSTATU(100)
BLDIES(100)	IAIRDFLG(100)	IOBSTOP(100)
BLGAS(100)	IAMOALF(4)	IOGCDS(100,2)
BROMU(100)	IBEYOND(100)	IOLDDATA(100)
CLAVGAS(100)	IDEGRAD(350)	IOLDMVDT(4,100)
CLDIES(100)	IDELAY(100)	IONBOARD(25)
CLGAS(100)	IDESTX(100)	IONROAD(100)
CPERC(50)	IDESTY(100)	IOPSTU(100)
CSLR(7,100)	IDETA(4,100)	IPRVENG(4,10)
DIESRAT(100)	IDETR(4,100)	IRAFT(4)
DNU(100,14)	IDETV(4,100)	IREDCON(100)
DPERS(100)	IDIESF(4)	IRFEBA(100)
DPERSU(100)	IDSTMOVL(100)	ISIZE(150)
EQINIT(100,14)	IFIREFA(100,25)	ISPTUNIT(150)
FIRED(100)	IFIRRFL(100,2)	ISTATU(100)
FRACTG(100)	IGADET(4,10)	ISTOP(100)
FRLKOPST(100)	IGASF(4)	ITCODE(100)
FTMVMNT(100)	IMDEADCP	ITEQU(100,14)
GASRATE(100)	INOG(100)	ITPPL(100)
ITRAV(100)	LOSPROB(100,25)	NXHGCM(150)
ITYPEU(100)	MENIN(2,100)	OBSDEL(100)
IUA(100)	MENNOW(2,100)	OLDFLG(3,100,14)
IUDEP(100)	MOUNTED(4)	PCSLR(50)

IUNIM(4)	MOUNTOLD(4)	PDIR(100, 2)
IUNIMXY(4)	MOVECC(100)	PERNVC(100,5,4)
IUNIT(100)	MSIZE(100)	PERS(100)
IUWID(100)	MUSLRA(100)	PERSI(100)
IVA(100)	MVDT1(100)	PERSWT(100)
IVAA(100)	MVDT2(100)	ROMU(100)
IVGSL(100)	MVDT3(100)	SIGENG(100)
IWT(100)	MVTCD(100)	SIGU(100)
IXY(100,2)	NAMOU(100)	SLOPET(100)
IXYIM(100)	NETAMU(100,14)	STT(50)
JOLDDATA(100)	NETAMX(100,14)	SWFU(100)
KLDPPL(50)	NETU(100)	TOTEQU(100,14)
KOLDDATA(100)	NFFLG(4)	TPCAS(50)
LDET(100)	NOWFLG(3,100,4)	UA(100)
LISTUN(150)	NTAMU(100,14)	UELEV(100)
LOSIND(100)	NTPEQDST(25)	UNNAME(3,150)
		USEEQU(100,14)
		VA(100)

b) Term Description:

All variables representing attributes which define a unit are listed below. Generally, the indices IU and JU are integers which range from 1 to 100 inclusively. However, there are instances in which the range may go from 1 to 150 inclusively. Indices other than IU or JU are usually dummy indices referencing packed data arrays.

BDIR (JU,J)	Sine and cosine of angle faced by unit JU during preceding timestep-cartesian coordinate system J=1 sine of angle of unit JU J=2 cosine of angle of unit JU BDIR is copied from PDIR (common BSTAT) each minute and is used by fore-ground display

BLAVGAS (IU)	Basic load of aviation fuel for unit IU.
BLDIES (IU)	Basic load (initial amount) of diesel fuel (in gallons) for unit IU)
BLGAS (IU)	Basic load (initial amount) of gas for unit IU in gallons.
BROMU (JU)	Rate of movement of unit JU in meters/min for the preceding timestep.
CLAVGAS (JU)	The current load of aviation fuel for unit JU
CLDIES (JU)	Current load (aviation amount) of diesel fuel for unit I in gallons.
CLGAS (JU)	Current load (present amount) of gas for each of unit JU in gallons
CPERC (JU)	The JUth half word of CPERC is used to control temporary casualty reports for unit JU resulting from reaching the personnel casualty threshold. The JUth half word is set equal to the current number of persons in unit JU when unit JU starts an engagement and when unit JU sends a temporary casualty report.
CSLR (JU,J)	CSLR is a half-word array of 14 by 100. Fourteen half-words (one per equip that each unit may have) is used for each unit to store the total equipment casualties incurred by unit JU since unit JU engages.
DIESRAT (JU)	The diesel fuel consumption rate for Unit JU in gallons per km. Based on equipment remaining in unit JU. This value is recalculated every time step in subroutine USEFUEL.
DNU (JU,J)	Casualties for equipment J in unit JU during current timestep.
DPERS (JU)	Personnel casualties for unit JU during current timestep.

DPERSU (JU)	Casualties for unit JU if unit JU is unsupressed during current timestep.
EQINIT (IU,J)	Initial amount of Jth equipment type in unit IU.
FIRED (JU)	= .TRUE. if ground unit JU has already fired its air defense weapons during current quarter minute.
FRACTG (IU)	Fraction of area of unit IU that is fired into when area fire is employed against unit IU.
FRLKOPST (IU)	The fraction of people who are lookouts for unit IU
FIMVMNT (JU)	The fraction of total personnel in unit JU and are on their feet. =0 If the whole unit is mounted =1. If the unit is dismounted completely
GASRATE (JU)	The gasoline consumption rate for unit JU in gallons per km. Based on total equipment remaining in Unit JU. This value is recalculated every time step in subroutine USEFUEL.
IADWUNIT(JU)	Bit flag indicating that a target unit has been hit with air-delivered ordnance (JU=1,4)
IAGDET (JU,JAU,JAS)	Bit matrix indicating whether ground unit IGU has already been detected by air unit JAU with sensor JAS. IGU is used to determine JU.
IAIRDFLG (IU)	Air defense flag for ground unit IU. =1,fire at will; =2, fire only if attacked; =3, do not fire.
IAMOALF (JU)	Ammunition request outstanding indicator JUth bit (most significant as bit I): =0 JUth unit does not have a low ammo request outstanding =1 JUth unit has a low ammo request outstanding. In this case unit JU will generate no additional ammo requests until unit is resupplied.

IBEYOND (JU) — A packed array (half-words) containing the following data for the JUth (1 < = JU < = 100) unit:

first half-word: X-coord. (divided by four) of destination point when traveling across an area obstacle

second half-word: Y-coord. (divided by four) of destination point when traveling across an area obstacle

IDEGRAD (JU) — Byte packed array containing the movement degradation factor expressed as an integer percent applied to the Jth equipment in unit K

The factor for the Jth (1 < = J < = 14) equipment of the Kth (1 < = K < = 100) unit is stored in the Ith byte from IDEGRAD(1), where

I=(K-1)*14+J-1

IDELAY (JU) — Delay counter (general purpose) for unit JU.

IDESTX (JU) — The X-coord. of the exit point out of an area obstacle for the JUth (1 < = JU < = 100) unit

IDESTY (JU) — The Y-coord. of the exit point out of an area obstacle for the JUth (1 < = JU < - 100) unit

IDETA (JU,J) — A bit-packed array containing aural detection verdicts. Unit J has a current detection of unit K if the Kth bit from IDETA(I,J) is set

IDETR (JU,J) — A bit-packed array containing ground radar detection verdicts. Unit J has a current detection of unit K if the Kth bit from IDETA(I,J) is set

IDETV (JU,J) — A bit-packed array containing visual detection verdicts. Unit J has a current detection of unit K if the Kth bit from IDETA(1,J) is set

IDIESF (JU) — Diesel resupply request outstanding indicator. Please see IAMOALF for interpretation.

IDSTMOVL (JU)	The distance that unit JU has moved since the last line of sight calculation.
IFIREFA (JU,J)	The percentage of unit K visually detected by unit JU, is stored in the Kth byte from IFIREFA(JU,1)
IFIRRFL (JU,J)	Indicators of fire received (J=1) and fire laid on (J=2) for unit JU ; CODE : 0 - Direct and indirect fire 1 - Indirect fire only 2 - Direct fire only 3 - No fire from direct or indirect weapons.
IGADET (JU,JAU)	Bit matrix indicating whether ground unit IGU has already detected air unit JAU. IGU is used to determine JU.
IGASF (JU)	Gasoline resupply request outstanding indicator for interpretation please see IAMOALF.
IMDEACP (JU)	The simulation time at which communications is to be restored to unit JU, if unit JU is a C+C HG which has lost communications because it has been destroyed
INOG (IU)	Operational grouping to which unit IU belongs. (If 0, unit IU does not belong to any op. group) If IU is an air unit, INOG = 1 is a red air unit and INDO = 2 is a blue air unit.
IOBALRT (JU)	A bit array used to communicate to subroutine LOWALRT whether or not an obstacle alert should be generated for a unit during the current timestep

IOBSTATU (JU) Status of the JUth (1 < = JU < = 100) unit with respect to obstacles:

= 0 unit is not stopped by an obstacle

= 1 unit is stopped at an obstacle, and engineering support is available

= 2 unit is traversing through an obstacle

= 3 unit is stopped at an obstacle waiting for engineering support

= 4 unit is stopped by an obstacle requiring the construction of a bridge without the aid of engineering support

= 5 unit is stopped by an obstacle requiring the construction of a bridge with the aid of engineering support

the construction of a bridge with the aid of engineering support

= 6 unit is stopped temporarily in order to prepare for a bridge crossing operation

IOBSTOP (JU) A packed array (half-words) containing the following data for the JUth (1 < = JU < = 100) unit:

first half-word: a flag indicating whether the JUth unit has arrived at a designated point beyond an area obstacle (0=NO,1=YES)

second half-word: the integer designation of the obstacle (1 thru 50) currently delaying the Ith unit

IOGDS (IU,J) Coordinates of planned location of unit IU relative to location of the forward most unit of its op. grouping is deployed. J=1 is the forward distance of unit from the location of the forward most unit (negative is rearward) : J=2 is the lateral distance from this location (plus is to the left).

IOLODATA(JU)	Half-word packed array containing movement data for the JUth (1 < = JU < = 100) unit where the first half word stores the operational state of the JUth unit and the second half-word stores the movement code of the JUth unit
IOLDMVDT (J,JU)	An array used to save the old movement data values for unit JU if it becomes engaged. These values may then be restored at the termination of the engagement, allowing unit then to continue as before. IOLDMVDT(1,JU) = MV1CD(JU) IOLDMVDT(2,JU) = MVDT1(JU) IOLDMVDT(3,JU) = MVDT2(JU) IOLDMVDT(4,JU) = MVDT3(JU)
IONBOARD (IU)	Byte packed array storing the number of the engineering unit that will provide a raft for the Kth (1 < = K < = 100) unit and will also escort the Kth unit through a given water obstacle where K is the Kth byte from IONBOARD(1);
IONROAD (JU)	Flag indicating whether the JUth (1 < = JU < = 100) unit is traveling along a road
IOPSTU (IU)	Operational state of unit IU. If IU is an air unit, IOPSTU = 1 if IU is on a recommaissance mission and = 2 if IU is on a strike mission.
IPRVENG (JU,JAU)	Bit matrix indicating whether ground unit IGU is currently engaging air unit JAU with air defense weapons. IGU is used to determine JU.
IRAFT (IU)	Bit packed array containing data indicating whether the Kth (1 < = K < = 100) unit has a raft in its equipment list where K is the Kth bit from IRAFT(1)
IREDCON (JU)	The readiness condition for unit JU

IRFEBA (JU)	Distance of unit JU from local enemy FEBA' (negative number implies that unit is behind enemy FEBA.
ISIZE (IU)	Unit size of unit IU. A negative sign preceding size means the unit is a command post. IU=1 Squad IU=2 Section IU=3 Platoon IU=4 Company/battery/troop IU=5 Battalion IU=6 Regiment IU=7 Brigade IU=8 Division IU=9 Corps IU=10 Army IU=11 Army group
ISPTUNIT (JU)	For unit JU, the support fire unit assigned to unit
ISTATU (IU)	Status code of unit IU: -1 -- air unit 1 -- in engagement, eligible for direct fire against enemy units in same engagement 0 -- other
ISTOP (JU)	The length of time which unit JU has been stopped
ITCODE (JU)	Target marker code indicating primary (=0) or secondary (=1) targets in UA (JU) (sometimes used as temporary storage) for unit JU
ITEQU (IU,J)	List of (J=1 to 14) equipment types in unit IU, principal type first and sink or catch-all type last. (Support-fire weapons are always listed before other categories.)

ITPPL (IU)	Unit arm/branch/duty. A negative sign preceding means unit is an observation post. 1 Infantry 2 Mech infantry 3 Airmobile infantry 4 Airborne infantry 5 Armor 6 Cavalry 7 Armored cavalry 8 Anti-tank 9 Artillery, towed 10 Artillery,SP 11 Air defense 12 Engineer 13 Signal 14 Maintenance 15 Medical 16 Ordnance 17 Quartermaster
ITRAV (IU)	Travel code of unit IU; 0 - not applicable 1 - unit avoids engagement if possible 2 - unit neither desires nor avoids engagement unit destination is reached 3 - unit seeks engagement 4 - air unit
ITYPEU (IU)	Unit type for unit IU
IUA (JU)	See definition of UA
IUDEP (IU)	Depth of unit IU in meters.
IUNIM (JU)	Bit-packed array indicating which units were fired on during the minute (1=fired on), where word 1,bit 0 is the indicator for Unit 1, etc.

IUNIMXY (JU)	Four word array of bit flags indicating if a unit has been hit with fire directed at an XY point; such units are then flagged not to have an impacting fire symbol at their center, since the XY-point symbol is within the area of the unit
IUNIT (IU)	Routing code for alerts generated by unit IU =0 Ignore (route to NOONE) 1 Route to controller 1 2 Route to controller 2 3 Route to controller 3 4 Route to controller 1 and 3 5 Route to controller 2 and 3 6 Route to controller 1 and 2 7 Route to controller 1,2 and 3
IUWID (IU)	Width of unit IU in meters.
IVA (JU)	See definition of VA
IVAA (JU)	This array is a temporary storage array for communication between the movement subroutines.
IVGSL (JU)	The vegetation (first half-word) and soil (second half-word) classes for unit JU.
IWT (JU)	Target allocation vector for unit JU. In a given weapon-target set, if unit is a weapon unit, IWT contains the number of enemy units at which it can fire at. If unit is a target unit, IWT contains the number of enemy units which can fire at it. Also, IWT is sometimes used as temporary storage.
IXY (IU,J)	Actual position coordinates (J=1 for X,J=2 for Y) of unit IU.

IXYIM (JU) — Each element of this array, if non-zero, indicates the XY location of an impacting fire, as follows:

-sign of the word indicates RED(-) or BLUE(+)
-after converting to a positive number,
 bits 0-15 -- X location
 bits 16-31 -- Y location

JOLDDATA (JU) — Half-word packed array containing movement data for the JUth (1 < = JU < = 100) unit where the first half word stores the first movement data value (see MVDT1(JU)) of the JUth unit and the second half word stores the second movement data (see MVDT2(JU)) of the JUth unit

KLDPPL (JU) — Number of personnel killed in unit K due to minefields where K is the Kth half-word from KLDPPL(1).

KOLDDATA (JU) — Half-word packed array containing movement data for the JUth (1 < = JU < = 100) unit where the first half-word stores the third movement data value (see MVDTB(JU)) of the JUth unit and the second half-word stores the distance to be traversed across the obstacle by the JUth unit

LDET (JU) — UGS detection flag for unit JU

Byte 0 =0 no detection unit JU by any UGS field
 =1 unit JU was detected by a UGS field
Byte 1,2,3 UGS field numbers which detected unit JU

LISTUN (IU) — A byte packed array of pointers used to control the order which units are listed in on all the menus and the status report. Each word corre-ponds with a unit IU. The word points to the next unit in the list after IU using the following method: each byte of the word contains the unit number of the next unit in the proper size list.

0 byte(left most) platoon list
1 byte(left middle) company list
2 byte(right middle) battalion and above list
3 byte(right most) list of all units

LOSIND (JU)	Byte packed data for each unit JU. Byte 0 is a flag that determines whether or not the unit has moved sufficiently to recompute lines of sight. If byte 0=1, the lines of sight must be recomputed. Bytes 1 and 2 are not used. Byte 3 is the terrain data grid block number for the unit.
LOSPROB (JU,J)	The array contains the percentage of unit K exposed to unit JU; where K is the Kth byte from LOSPROB(JU,1)
MENIN (2,IU)	A half-word packed array which contains the initial amount of the four types of personnel in each unit IU: Left half-word (1,IU)=no. of CO in unit IU Right half-word (I,IU)=no. of OFF in unit IU Left half-word (2,IU)=no. of EMLDR in unit IU Right half-word (2,IU)=no. of EM in unit IU
MENNOW (2,IU)	A half-word packed array which contains the current amount of the four types of personnel in each unit IU: Left half-word (1,IU=no. of CO in unit IU Right half-word (1,IU)=no. of OFF in unit IU Left half-word (2,IU)=no. of EMLDR in unit IU Right half-word (2,IU)=no. of EM in unit IU
MOUNTED (4)	Bit-packed array giving mounted/dismounted status of each unit. Zero means mounted, one means dismounted.
MOUNTOLD (4)	Bit-packed array giving mounted/dismounted status of each unit during previous timestep.
MOVECC (JU)	Indicator of interactive movement CMD. and control =0 unit is not under interactive control =1 unit is under interactive control (mounted) =2 unit is under interactive control(dismounted)

MSIZE (IU)	Size of unit IU (like ISIZE was intended to be). Should reflect the actual number of personnel in the unit. 1 = Squad 2 = Section 3 = Platoon 4 = Company/battery/troop 5 = Battalion 6 = Regiment 7 = Brigade 8 = Division
MUSLRA (IU)	Distance of unit IU to enemy FEBA or nearest enemy unit;used to select proper mode distribution vector.
MVDT1 (IU)	Movement data for unit IU
MVDT2 (IU)	Movement data for unit IU
MVDT3 (IU)	Movement data for unit IU
MVTCD (IU)	Movement code of unit IU: 1-normally engaged 2-withdrawing 3-deploying(not in position) 4-deploying(in position waiting for other units) 5-moving in fixed direction 6-moving along route 7-halted 8-moving toward fixed point 9-moving toward point relative to friendly operational grouping 10-moving toward point relative to enemy operational grouping 11-moving toward point relative to friendly engagement FEBA 12-moving toward point relative to enemy engagement FEBA

	13-moving toward point relative to friendly unit
	14-moving toward point relative to enemy unit
	15-deploying while not engaged(not in position)
	16-deployed while not engaged (in position waiting for other units).
NAMOU (IU)	Number of ammunition types in unit IU
NETAMU (IU,J)	Current amount in tenths of a round, of Jth ammo type carried by unit IU. Actual ammo type given by NTAMU(IU,J)
NETAMX (IU,J)	Initial amount in tenths of a round, of Jth ammo type carried by unit IU. Actual ammo type given by NTAMU(IU,J).
NETU (IU)	Number of equipment types in unit IU
NFFLG (JU)	Firing array - using for detection subroutine the Kth bit from NFFLG(1) indicates whether the Kth unit is firing or not (=1 for firing or not (=1 for firing)
NOWFLG (I,JU,K)	Receiving fire indicator table for current time-step NOWLFG is a bit table indexed. NOWFLG is interpreted similarly to OLDFLG.
NTAMU (IU,J)	Ammunition type number of Jth ammo type carried by unit IU.
NTPEQDST (JU)	Equipment class destroyed in unit K by minefield where K is the Kth byte from NTPEQDST(1)
NXHGCM (IU)	The unit number of the unit which is next higher command for unit IU(IU=1,150)

OLDFLG(I,JU,K) Receiving fire indicator table for last timestep:

OLDFLG is a bit table:

First subscript = type of fire

1=direct fire (small arms)

2=indirect fire (mortars)

3=support fire (artillery)

Second subscript = unit number of unit receiving fire

Third subscript = word number of firing units

OLDFLG is referenced as follows:

I=type of fire	IFU=firing unit
JU=receiving unit	K=(IFU+31)/32 word number
IB=MOD(IFU,32) bit number for IFU	K=(IFU+31)/32 word number for unit IFU

Then:

the IBth bit (most significant bit is bit 0) of OLDFLG(I,JU,K)

=0 means unit IFU fired fire type I at unit JU

=1 means unit IFU did not fire fire type I at JU

OBSDEL (IU) Obstacle delay counter for unit IU

PCSLR (JU) The Ith half-word is a counter of the accumulated personnel casualties of unit I since shelling starts or since last temporary casualty report.

PDIR (IU,J) Direction faced by unit IU - stores as SIN(J=1), COS(J=2). Cartesian coordinate system used.

PERNVC (IU,IPVC,K) Number of personnel in each vulnerability class IPVC for uni- IU where K=:

1-actual number in previous time-step

2-actual number in current time-step'

3-unsuppressed number in previous time-step

4-unsuppressed number in current time-step

PERS (IU)	Number of personnel currently in unit IU
PERSI (IU)	Initial number of personnel in unit IU
PERSWT (JU)	Weight of unit JU as a personnel target.
ROMU (JU)	Rate of movement of unit JU in meters per minute.
SIGENG (JU)	Total energy expended since the start of the simulation by a single person in the JUth unit; this number is used to compute an overall human degradation effect for the JUth unit
SIGU (JU)	Suppression factor: fraction of men in unit JU who are suppressed.
SLOPET (JU)	The instantaneous terrain slope for unit JU
STT (JU)	The Ith half-word of STT stores time in minutes at which firing of unit I starts. -1 is stored if unit I has not started an engagement.
SWFU (JU)	Area support fire weapon rounds/unit time received by unit JU
TOTEQU (IU,J)	Total number of pieces remaining of Jth equipment type carried by unit IU.
TPCAS (JU)	The JUth half-word is a counter of the total personnel casualties for unit JU since firing starts
UA (JU)	Total weighted target value of unit JU (sometimes used as temporary storage).
UELEV (JU)	The elevation of unit JU.
UNNAME (3,IU)	12 character name associated with each of IU(1-150) units
USEEQU (JU,J)	Number of pieces manned for the Jth equipment type carried by unit JU.

VA (JU)	Weighted value of target elements detected in unit JU (sometimes used as temporary storage).

Because units are acted upon by almost every module, a list of how each of how each of the unit variables gets changed would constitute a report volume in and of itself. Thus, when the user becomes interested in one of these many variables, the super index listing should be consulted along with write-ups on the particular module and other applicable term definitions. The unit variables which must be initialized by a user (data base input variables) are listed below, along with the input deck in which they are defined:

EQINIT (100,14)	Unit input deck
FRACTG (100)	Unit input deck
FRLKOPST (100)	Namelist input deck
INOG (100)	Unit input deck
IOPSTU (100)	Unit input deck
ISIZE (150)	Unit input deck
ITEQU (100,14)	Unit input deck
ITPPL (100)	Unit input deck
ITRAV (100)	Unit input deck
ITYPEU (100)	Unit input deck
IUDEP (100)	Unit input deck
IUNIT (100)	Namelist input deck
IUWID (100)	Namelist input deck
IXY (100,2)	Namelist input deck
LISTUN (150)	Deck to control listing of units and next higher command
MENIN (2,100)	Unit input deck
MENNOW (2,100)	Unit input deck
MSIZE (100)	Unit input deck
MVDT1 (100)	Unit input deck
MVDT2 (100)	Unit input deck
MVDT3 (100)	Unit input deck
MVTCD (100)	Unit input deck
NAMOU (100)	Unit input deck

NETAMU (100,14)	Unit input deck
NETU (100)	Unit input deck
NTAMU (100,14)	Unit input deck
NXHGCM (150)	Deck to control listing of units and next higher command
OBSDEL (100)	Unit input deck
PDIR (100,2)	Unit input deck
SIGU (100)	Unit input deck
TOTEQU (100,14)	Unit input deck
UNNAME (3,150)	Unit input deck

c) Term References:

The arrays containing data describing unit attributes are indexed by unit number (integer) IU (or JU), where 1 ≤ IU,JU ≤ 100. Arrays with the unit number index IU indicate that initial values via input must be entered into these arrays. The index JU indicates that the arrays contain data that is computed by the math model (rather than entered by input). Note that many arrays contain packed data; in this case, the index does not correspond to unit number (but ultimately, data is retrieved based upon unit number and index number). The above arrays are used by one or more of the following subroutines:

ACTIVATE	AIRMOV	CK4XING	DTECTPRB
ADD2LIST	AIRMOV2	CLEARDET	EFFNS
ADFALERT	AMMOLINE	CLSPTG	ENAREA
ADJDIR	AMOVUL	CMAIN	ENGRSPT
ADJROM	ANYFOE	CMDUNINP	ENGRUNIT
ADW	AREA	CMSEGMNT	ENGUPDAT
ADWALERT	ARRIVE	CNTRLMST	ENTCR
AIRABORT	AURAL	CRDLIC	EQIPNT
AIRCAS	BDTGTS	DALERT	EQUPLINE
AIRDEVENT	BULDBROG	DEPLOC	FINDFO
AIRERROR	CBIFIR	DEPLOY	FINDWA
AIREVENT	CBTVAL	DETECT	FINFUN
AIRGRND	CHGCRT	DIRMOV	FIRAGEN
AIRHOTZN	CHGOPN	DLOSINP	FIRALO

FIRELG
FIREVNT
FIRSORT
FIXLIST
FIXOGCDS
FORMAIN
FORRAM
FORSIGI
FRNTGS
FSTOT
FUELLINE
FWDLIN
GENFIR
GNSPTG
HUMAN
INIT
INITELEV
INPUT
ISDEADCO
JOINOG
LATDST
LEAVEOG
LOSCOMP
LOSINP
LOSVEG
LOWALRT
MANEUVER
MANNING
MKUNLIST
MOVE
MOVMNT
NEWENG
NEWFWDUN
NEWMOV
NGARGN
NOTGT
NXUNIT
OBSDELAY
OBSTACLE
OBSUPDAT
OGCENTER
OGDIR
OGFRNT
OGHFRONT
OPINP
OGLOC
OGLOC2
OGTYPE
OG2UNI
OPPLAN
ORDPRI
ORGFIR
OTHRDMG
OVERUN
PARE
PERSLINE
POINTGT
PREMOV
PRNTFIR
RADAR
RAMGEN
RDSON
REDCON
REDIST
REL2FWDU
RESUPPLY
RMVOPG
ROADCHK
SAVEINP
SAVEOLD
SCHMU
SCHRMU
SETFLG
SETIMP1
SETIMP2
SETOGCDS
SOIL
SPTALO
SPTFIR
STATREP
STATREP1
STEP
STLKUP
SUPRES
TASKORG
TGTCAT
TGTLIST
TGTORD
UASCK
UNBLOK
UNINP
UNITLINE
UN2FEB
USEFUEL
UTINP
VISUAL
WPNEFF
WPNFIR
WPVEL
WTHDRW
WTSETS
WTSUB

3.149 UNIT CLASS

3.149.1 English-language Definition

a) Term Name:

Unit Class

b) Term Description:

The English-language description of "Unit Class" is identical to the term "Vulnerability to Air Ordnance." See the write-up of that term.

c) Term References:

The English-language term references for "Unit Class" are identical to "Vulnerability to Air Ordnance."

3.149.2 Programmatical Description

a) Term Name:

IUCLS(IUT)

b) Term Description:

The Programmatical description of "Unit Class" is identical to the term "Vulnerability to Air Ordnance." See the write-up of that term. IUCLS(IUT) is a data base input variable that is defined in the namelist input deck (read in using variable IAIRVUL).

c) Term References:

The Programmatical term references for "Unit Class" are identical to "Vulnerability to Air Ordnance."

3.150 UNIT LOCATION

3.150.1 English-language Definition

a) Term Name:

Unit Location

b) Term Description:

Unit location for ground units is represented by an X,Y point. This point is considered to be the center of mass of the unit. Although all ground units have a width and depth and are graphic displayed as rectangles, the unit is considered to be located at a single X,Y point. All references to unit location for purposes of terrain interaction, detection, fire, movement, etc., use this point.

c) Term References:

Unit location is input via the data base at initialization. This position may be changed via command and control action prior to commencing the running of the model. Thereafter, unit locations are updated by the movement function each time step.

3.150.2 Programmatical Definition

a) Term Name:

IXY(IU,J)

b) Term Description:

IXY (IU,J)	Actual position coordinates (J=1 for X,J=2 for Y) of unit IU in meters.

Ground unit positions are updated once each time step.

c) Term References:

IXY (IU,J) is a unit attribute indexed on unit number, ($1 \leq IU \leq 100$); J=1 indicates X-position, J=2 indicates Y-position. Some 70 subroutines use this array.

IXY(IU,J) is a data base input variable, an interactively generated variable, as well as a model generated variable. As a data base input variable, it is defined in the unit input deck; as an interactively generated variable, it is created from the unit location menu, maneuver menu, and tasking organization menu; and, as a model generated variable, it is used in subroutine ADW (air units only), AIREVENT (air units only), AIRMOV (air units only), CRDLIC (no menu was ever written which would cause this subroutine to be called), DIRMOV, MOVMNT, OBSUPDAT, OTHRDMG (air units only), and OVERUN.

3.151 UNIT SIZE

3.151.1 English-language Definition

a) Term Name:

Unit Size

b) Term Description:

Unit size is used primarily by the graphic display programs to determine the appropriate tactical overview symbol to be displayed for the unit. It is also used by the command/control function to determine which units are to be displayed when one of the buttons "platoon", "company" or "battalion" is depressed.

c) Term References:

The foreground graphics display programs and command/control programs are the principal users of the size of each unit. Size is a unit attribute.

3.151.2 *Programmatical Definition*

a) Term Name:

ISIZE(IU)

b) Term Description:

Array ISIZE is both a data base input variable and an interactively generated variable. As a data base input variable, it is defined in the unit input deck for units 1-100, the operational grouping input deck for units 101-120, and the higher and adjacent unit input deck for units 121-150; as an interactively generated variable, it is created from the tasking organization menu for units 101-120. It is defined as follows:

ISIZE (I) Unit size of unit I. A negative sign preceding size means the unit is a command post.

I=1 Squad
I=2 Section
I=3 Platoon
I=4 Company/battery/troop
I=5 Battalion
I=6 Regiment
I=7 Brigade
I=8 Division
I=9 Corps
I=10 Army
I=11 Army group

Array ISIZE(IU) is initialized as part of the data base inputs and not changed for units (IU index lessthan 100). For operational grouping I, where I is in the range ($1 \leq I \leq 20$), ISIZE(I+100) represents the size of the op group. The op group value for ISIZE is determined when the op group is defined, which is either through system initialization or through the task organization menu. For adjacent units having unit numbers in the range ($121 \leq IU \leq 150$) ISIZE represents the size of up to 15 red adjacent units, and up to 15 blue, respectively.

c) Term References:

Unit number (or op group/adjacent unit number) is used as the reference into the ISIZE array to identify the unit (or op group/ adjacent unit) size for purposes of graphic display. The index, I, ranges from ($1 \leq I \leq 100$) for normal units, ($101 \leq I \leq 120$) for op groups, and ($121 \leq I \leq 150$) for red and blue adjacent units. The following subroutines use array ISIZE:

CMDUNINP	OGINP
FIRAGEN	OGLOC
FIXLIST	STEP
FORRAM	TASKORG
LEAVEOG	UNINP
NEWFWDUN	USEFUEL

3.152 UNIT STATUS

3.152.1 English-language Definition

a) Term Name:

Unit Status

b) Term Description:

The unit status flag is used to indicate

1) those units that are active in the exercise being run
2) among active units, those involved in engagements and not removed from combat status
3) in conjunction with travel code, which units are active air units.

All active units are processed and displayed. Direct fire allocation processing is performed on active combat units in existing engagements, whereas non-combat units are not eligible for direct fire allocation. Air units are assigned on inactive ground status to eliminate them from ground unit processing.

c) Term References:

Unit status is a unit attribute. The casualty accounting function determines when a unit becomes inactive due to a total loss of personnel. The engagement function determines which of the active units involved in an engagement are eligible for direct fire.

3.152.2 Programmatical Definition

a) Term Name:

ISTATU(IU)

b) Term Description:

ISTATU (IU) — Status code of ground unit IU:

-1 -- defunct

1 -- in engagement, eligible for direct fire against enemy units in same engagement.

0 -- other

Status code of air units = - 1

ISTATU (IU) is set to -1 when all personnel are lost in a unit and that unit remains inactive for the remainder of the exercise. Each time step, every existing engagement is proce-sed to determine which units can be designated an ISTATU(IU) = 1 for that engagement. All air units are assigned ISTATU(IU) = -1.

ISTATU(IU) is both an interactively generated variable and a model generated variable. As an interactively generated variable, it is created from the activate menu, and, as a model generated variable, it is used in subroutines ANYFOE, FWDLIN, INIT, and STEP.

c) Term References:

Unit status is a unit attribute indexed by unit number, where ($1 \leq IU \leq 100$). Status for air units is stored in ISTATU (27) through ISTATU (36), inclusive. The array ISTATU is used by some 65 subroutines.

3.153 UNIT TYPE

3.153.1 English-language Definition

a) Term Name:

Unit Type

b) Term Description:

Unit type is a classification of each unit according to its general type. Two units of the same type need not have the same size, organization, or equipment. They do share a number of attributes which are defined in the programmatic description. Eleven different unit types are defined in the data base:

mechanized infantry
armor
scout
anti-tank
light mortar
heavy mortar
artillery
air defense
combat engineer
combat trains
headquarters

c) Term References:

The term unit type itself is an attribute of units in the model. A large number of attributes combine to define a particular unit type. Unit type is used in all major program modules.

3.153.2 Programmatical Definition

a) Term Name:

DISMAX(IUT)
HU(IUT)
HUN(IUT)
IAIRVUL(IUT)

IDPRD(IUT,ICOLOR)
IPRNO(IUT,J)
IPWT(IUT,J)
ITAP(IUT,ICOLOR)
ITYPDW(IUT,ICOLOR)
KEQMOV(IUT,ICOLOR)
MAXID(IUT,ICOLOR)
MFIGHT(IUT,ICOLOR)
NGARNG(IUT,ICOLOR)
POSFAC(IUT,ICOLOR,IFTYP)
RTGT(IUT)
TYPFAC(IUT,ICOLOR)
WU(IUT)
WUN(IUT)

b) Term Description:

The following program variables are input at initialization and combine to define unit type:

DISMAX (IUT)

The maxium distance for the line of sight calculation for unit type IUT.

HU (IUT) — The effective height for the type IUT unit for visual detection

HUN (IUT) — Height in meters of a single element of a unit of type I - for line of sight calculations (1 < = IUT < = 20)

IAIRVUL (IUT) — An indicator describing the vulnerability of unit type IUT to air ordnance

0 = not used
1 = not armored
2 = light armored
3 = heavily armored

IDPRD (IUT,ICOLOR) For each unit type IUT, red(ICOLOR=1), or blue (ICOLOR=2), code that indicates which range to use with target-acquisition probability curve to determine acquisition of targets for indirect-fire weapons in unit
0-implies distance from firing unit to target unit
1-implies distance from local friendly FEBA to target unit.

IPRNO (IUT,J) Byte-packed array giving 40 equipment manning priorities for unit type IUT. Numbers of corresponding equipment types given by array input.

IPWT (IUT,J) For given unit type IUT, a byte packed list of equipment type numbers. Their associated manning priorities are given by corresponding bytes of IPRNO.

ITAP (IUT,ICOLOR) Number of the target-acquisition probability curve to use for detection of units by a ground-based surveillance system. For IUT = type of unit to be detected; ICOLOR = 1 if unit is red, = 2 if unit is blue. If ITAP = 0, detection probability = 1.0

ITYPDW (IUT,ICOLOR)
Maximum distance beyond end of enemy FEBA in an established engagement that an unengaged red (ICOLOR=1) or blue(ICOLOR=2) unit (or its operational grouping) of a given type IUT will be allowed to join existing engagement (rather than form a new one.) This distance applies only to encounters between units, and distance may be negative.

KEQMOV (IUT,ICOLOR)
For each unit type IUT,Red(ICOLOR=1) or blue (ICOLOR=2), a code indicating which types of equipment determine rate of movement for the unit:

0 - all equipment

1 - direct-fire weapons only

2 - direct-and indirect-fire weapons only

3 - support-fire weapons only

4 - nonweapons only

MAXID (IUT,ICOLOR) For each unit type IUT, red(ICOLOR=1) or blue (ICOLOR=2), maximum distance forward of friendly FEBA that its indirect fire weapons will fire against targets in same engagement, when target units are not eligible to receive direct fire. This distance is a constraint only for weapon units against target units in same engagement.

MFIGHT (IUT,ICOLOR)
For each unit type IUT, red(ICOLOR=1) or blue (ICOLOR=2), range at which a unit must initiate an engagement with an enemy unit.

NGARNG (IUT,ICOLOR)
For each unit type IUT, Red(ICOLOR=1) or blue (ICOLOR=2), range at which a unit is eligible to initiate an engagement with an enemy unit.

POSFAC (IUT,ICOLOR,IFTYP)
If a target unit of type IUT and color ICOLOR (1 = Red, 2 = Blue) is in special-threat category, and a weapon in category IFTYP (=KFICTR) is to be allocated, then its target weight is multiplied by POSFAC (IUT,ICOLOR,IFTYP).

RTGT (IUT) The reflectance for the type IUT unit

TYPFAC (IUT,ICOLOR)
Weighting factor for unit type IUT, red(ICOLOR = 1) or blue(ICOLOR = 2), that expresses importance of unit as a support fire target.

WU (IUT) — The effective width for the type IUT unit for visual detection

WUN (IUT) — Width in meters of a single element of a unit of type IUT for line of sight calculations

The above variables combine to define eleven distinct unit types in the model at this time:

1 = mechanized infantry
2 = armor
3 = scout
4 = anti-tank
5 = light mortar
6 = heavy mortar
7 = artillery
8 = air defense
9 = combat engineer
10 = combat trains
11 = headquarters

DISMAX and IAIRVUL are data base input variables that are defined in the namelist input deck.

HUN and WUN are model generated variables that are used in the main program CMAIN (data statement).

All other variables are data base input variables that are defined in the unit type input deck.

c) Term References:

The attributes which combine to define a given unit type, and which are listed above, are indexed by unit type number, IUT, where ($1 \leq IUT \leq 20$). Although a maximum of twenty types are allowed, only eleven are defined at the present time.

3.154 VEGETATION CLASS

3.154.1 English-language Definition

a) Term Name:

Vegetation Class

b) Term Description:

A vegetation class is defined by four types of features. These feature types are:

1) plots of grass
2) clumps of brush
3) tree trunks
4) crowns of trees

Each feature type is modeled as a set of objects having the geometry of solid right circular cylinders. The cylinders are of constant height and width for a given feature type. Furthermore, the centers of the cylinder objects are distributed randomly with a spatial Poisson distribution of a given density. In short, each feature type used to represent a vegetation class is characterized by sets of cylindrical objects having user defined heights, widths, and densities. Currently sixteen different classes of vegetation can be established concurrently. Each class is identified by an integer between one and sixteen inclusively.

c) Term References:

Vegetation class is referenced by class number. The data defining vegetation class is used primarily in the line of sight calculations and in the determination of unit movement rates. The target acquisition module, which yields various detection verdicts from various sensor and detection devices, makes use of vegetation data.

3.154.2 Programmatical Definition

a) Term Name:

H(16,4)RHO(16,4),W(16,4)

b) Term Description:

Vegetation classes are defined by the above three floating point arrays.

H(IVEG,J = Height in meters of the Jth ($1 \leq J \leq 4$) feature type in the IVEGth ($1 \leq IVEG \leq 16$) vegetation class

RHO(IVEG,J) = density in objects per twenty-five hundred square meters of the Jth ($1 \leq J \leq 4$) feature type in the IVEGth ($1 \leq IVEG \leq 16$) vegetation class

W(IVEG,J) = width (i.e., diameter) in meters of the Jth ($1 \leq J \leq 4$) feature type in the IVEGth ($1 \leq IVEG \leq 16$) vegetation class

These arrays are data base input variables that are defined in the data base disk file VEGCOMP in the DA area of the disk. They are initialized by input and remain constant throughout the simulation.

c) Term References:

The above arrays are indexed by vegetation class IVEG, where IVEG ranges from one to sixteen inclusively. Within vegetation class, each feature is indexed by feature type J, where J ranges from one to four inclusively.

3.155 VEGETATION POLYGON

3.155.1 English-language Definition

a) Term Name:

Vegetation Polygon

b) Term Description:

Vegetation polygon is either a triangle, rectangle, or circle used to define a region having a vegetation class different from that of the dominant class in the area of operation. The vegetation is assumed to be the dominant vegetation class unless vegetation polygons are used to trace out regions having non-dominant classes of vegetation. These polygons must be disjoint. Currently a maximum of 225 polygons can be used to trace out non-dominant vegetation regions. These polygons must be pre-defined via user input.

c) Term References:

Each vegetation polygon is referenced by an integer number between 1 and 225 inclusively. Vegetation polygons are used by the environmental/terrain logic to identify the surrounding vegetation features at each unit location. Knowledge of local vegetation class is required in order to retrieve data for line of sight calculations, derivation of unit movement rates, and computation of detection verdicts.

3.155.2 Programmatical Definition

a) Term Name:

ICL(225,),ITRC(225)XPOLY(5,225),YPOLY(5,225)

b) Term Description:

The arrays above are data base input variables that are defined in the data base disk file VEGLOC in the DA area of the disk. They contain data defining the attributes for vegetation polygon. Each polygon has associated with it a code for vegetation class, a code for polygon shape, and data defining the polygon. This data is input defined and remains constant throughout the simulation.

ICL (IVPOL) Vegetation class for vegetation polygon IVPOL

ITRC (IVPOL) Polygon type for each vegetation polygon IVPOL
= 1 for triangle
= 2 for rectangle
= 3 for circle

XPOLY (I,IVPOL) X - coordinates in meters defining each vegetation polygon. For triangular polygon L (XPOLY (1,L), YPOLY(1,L)), (XPOLY(2,L), YPOLY(2,L)) and (XPOLY(3,L), YPOLY(3,L)) are the three vertices and XPOLY(3,L) is the length of the polygon in the other dimension. For circular polygon (XPOLY(1,L) YPOLY(1,L)) is the center and XPOLY(3,L) is the radius.

XPOLY (I,IVPOL) Y - coordinate in meters defining each vegetation polygon. (See definition of XPOLY)

c) Term References:

Vegetation polygon is indexed by polygon number IVPOL, where IVPOL ranges from 1 to 225 inclusively. Vegetation polygon arrays are referenced by the following subroutines:

LINOBS
LOSINP
MICSOL

3.156 VISIBILITY (METEOROLOGICAL)

3.156.1 English-language Definition

a) Term Name:

Visibility (Meteorological)

b) Term Description:

Meteorological visibility refers to the degree of clearness of the atmosphere. The degree of clearness is measured in terms of distance, and is usually defined to be the greatest distance toward the horizon that prominent objects can be identified visually with the naked eye, under ideal ambient light level. Note that meteorological visibility can be excellent at night due to clear atmosphere, even though objects can not be visually detected (because of low ambient light level at night). Meteorological visibility is reduced due to presence of tiny particles in the atmosphere (i.e., fog, dust, rain).

c) Term References:

Meteorological visibility is referenced mainly by the weather module. In fact, the degree of clearness, measured in meters, is one of the characteristics which distinguishes the eleven different classes of weather (see term definition of weather class). Visibility is also used by the target acquisition (detection) module to determine visual detection verdicts. It is also used by the air module to determine whether it is safe to conduct air missions for various aircrafts. Finally, the movement logic considers visibility when computing environmental degradation factors to show unit movement.

3.156.2 Programmatical Definition

a) Term Name:

VISM

b) Term Description:

The floating point FORTRAN variable VISM stores the distance representing the degree of clearness of the atmosphere. This distance is measured in meters. The value stored in VISM may change as global weather changes during the simulation.

VISM is both an interactively generated variable and a model generated variable. As an interactively generated variable, VISM is created from the weather menu, and, as a model generated variable, it is used in subroutine INPUT.

c) Term References:

The FORTRAN variable VISM is referenced by including the common block WEATHR in the subroutine. This common block contains current global weather information. VISM is set initially by subroutine INPUT and then is changed whenever global weather is changed interactively (subroutine WETHRC). VISM is used in the following subroutines:

ADJROM	GRNDAIR
AIREVENT	INPUT
AIRMOV	VISUAL
DETECT	WETHRC

3.157 VULNERABILITY TO AIR ORDNANCE

3.157.1 English-language Definition

a) Term Name:

Vulnerability to Air Ordnance

b) Term Description:

When an air strike is created with more than one ordnance onboard the aircraft, a determination must be made as to which ordnance should be used on the selected target. There are six ordnance priority lists defined in the input data, three for area target, road, and bridge, and three for hard targets (unit). For a given target type (e.g., bridge) these ordnance priorities are numbers assigned to each ordnance type representing which ordnance is most effective against that target type. If an aircraft is equipped with more than one ordnance, the ordnance with the highest priority is used. In the case of hard target (units), the priority list to be examined is a function of unit type. Data is input at system start-up indicating a unit type's vulnerability to air ordnance. The designation is one of the following categories: not armored; light armored; heavily armored. This designation identifies the priority list to be examined for selection of ordnance.

c) Term References:

The air mission generation function uses vulnerability to air ordnance to determine which of possibly several air ordnances to use against a specified target. The vulnerability designation is an attribute of ground unit type.

3.157.2 Programmatical Definition

a) Term Name:

IAIRVUL(IUT),NPTE(IEQ,J),NSTE(IEQ,J)

b) Term Description:

Array IAIRVUL, sometimes identified as IUCLS in the code, contains an indicator for unit type vulnerability to air ordnance, defined as follows:

1 = not armored

2 = light armored

3 = heavy armored

This designation identifies whether the first, second, or third unit priority list is to be examined for ordnance selection. Ordnance priorities against units are stored in NPTE(IEQ,J) for equipment IEQ, when IEQ is an air ordnance. Array NSTE (IEQ,J) contains priorities for area target, road, and bridge, respectively for air ordnance. Values for NPTE and NSTE range from 0 to 160 in increments of 10, where 160 represents the highest possible priority.

IAIRVUL(IUT), NPTE(IEQ,J), and NSTE(IEQ,J) are data base input variables. IAIRVUL(IUT) is defined in the namelist input deck, and NPTE(IEQ,J) and NSTE(IEQ,J) are defined in the equipment input deck.

c) Term References:

Subroutine ORDPRI is the sole user of array IAIRVUL. IAIRVUL(IUT) has unit type as its index, where ($1 \leq$ IUT ≤ 20). ORDPRI is also the sole user of NPTE and NSTE for <u>air</u> ordnance, although these arrays are used in the ground fire module. Arrays NPTE(IEQ,J) and NSTE(IEQ,J) have equipment number as the first index, where $1 \leq$ IEQ ≤ 80, and the designator described above as the second index, where ($1 \leq J \leq 3$).

3.158 WEAPON

3.158.1 English-language Definition

a) Term Name:

Weapon

b) Term Description:

Equipment types having an equipment code greater than zero are ground weapons. There are four categories of firing weapons: direct fire, indirect fire, support fire, air defense. This code is used as a flag by the firing routines that process the four different weapon categories differently. All attributes describing an equipment pertain to a weapon and are discussed under the term "equipment". In particular, firing rate, ammunition type, weapons effectiveness data, etc., are essential for weapons, as opposed to equipments in general.

c) Term References:

Equipment code is obviously an equipment attribute used primarily to distinguish among non-firing and firing equipments (weapons), and, within firing equipments, the weapon category. This enables separate processing for each weapon type in the processing of the firing function.

3.158.2 Programmatical Definition

a) Term Name:

IEQCOD(IEQ)

b) Term Description:

IEQCOD (IEQ) — Equipment category code of given equipment Type IEQ:

-3-air sensor
-2-air ordnance
-1-aircraft
0-not a weapon
1-direct fire weapon
2-indirect fire weapon
3-support fire weapon
4-air defense weapon

IEQCOD(IEQ) is a data base input variable that is defined in the equipment input deck.

Any equipment, IEQ, for which IEQCOD(IEQ) > 0 is a ground fire weapon with appropriate defined for firing rates (ROFE(IEQ,J), ammunition type (IAMTE(IEQ,J)), weapons effectiveness data (IFNTB(I,J)) etc., defined for it. See programmatical description of "equipment" for a definition of these variables.

c) Term References:

IEQCOD(IEQ) is indexed by equipment, where ($1 \leq IEQ \leq 80$). It is used extensively throughout the math model to distinguish among equipment and weapon categories for various processing. In particular, ground fire subroutines FIRALO, SPTALO, and AIRCAS use IEQCOD to filter out direct and indirect fire weapons, support fire weapons, and air defense weapons, respectively for separate processing. After initialization, equipment code is not changed.

3.159 WEAPON EFFECTS COEFFICIENT SET

3.159.1 English-language Definition

a) Term Name:

Weapon Effects Coefficient Set

b) Term Description:

The weapons effects coe-ficient sets are the sets of coefficients input at system initialization to be used with the seven weapons effects functions. A given set of coefficients is to be used with a specified function for a given weapon firing at a particular target element (mode of fire and target unit operational state are other factors than can determine the coefficient set, but are currently not used -- see discussion of weapon effects and coefficient set tables in the next section).

c) Term References:

Weapon effects coefficient sets, as currently defined, are referenced by firing weapon and target element type. The Ground Fire Module uses this data exclusively.

3.159.2 Programmatical Definition

a) Term Name:

EFFDAT(I,J)

b) Term Description:

Array EFFDAT(I,J) is a data base input variable that is defined in the weapon effects deck. It is defined as follows:

EFFDAT (I,J)	Groupings of four constants (J) to be used with the Ith effects functions.

Data for this array is input at system initialization and never altered. The coefficient values have specific meanings depending on which function they are defined for.

c) Term References:

Array EFFDAT(I,J) is dimensioned , where I is the pointer from the Weapon effects table match, and J is a dummy index ranging from $(1 \leq J \leq 4)$. Subroutine EFFNS, which evaluatesthe seven functions describing ground fire effects, is the sole user of this data.

3.160 WEAPON EFFECTS FUNCTION

3.160.1 English-language Definition

a) Term Name:

Weapon Effects Function

b) Term Description:

The weapons effects function to be used for a ground fire weapon or air defense weapon against a ground target or an aircraft are determined by finding a match in the Weapons Effects Table on weapon number and target element number (either personnel vulnerability class or equipment). This match specifies the weapons effects function (and coefficient set) to be used.

Ground-Ground Weapon Effects Functions

Six ground-ground effects functions exist in the model, four of which are used. Those four are defined as follows:

Function 2 - Aimed Fire (used against personnel and equipment)

$$E_k = V * (a + bR + cR^2)$$

Where a,b,c are coefficients of a piece-wise quadratic fit to empirical casualty data as a function of range for various target elements.

E_k = (number of rounds) * (probability of kill for single round)

Function 3 - Area Fire (used against personnel and equipment)

$$E_k = VNL/A$$

$$= \frac{\text{(\# rounds) * (\# target elements in unit) * (lethal area of a round)}}{\text{(area of target being fired at}}$$

[L is approximated by a quadratic a + bR + cR^2 so that L is a function of range to the target unit]

Function 5 - Area Fire (used against personnel and equipment)

$$E_k = N * \left[1 - e^{(-VL/A)}\right]$$

N = # of target elements in target unit

V = # rounds fired

L = lethal area of each round

- approximated by quadratic $L = a + bR + cR^2$ so that L is a function of range to the target unit

A = area of target unit being fired into

Function 6 - Area Fire (used against personnel and equipment)

$$E_k = N * (- e)^{-VLF_x/A}$$

Same as function 5 except

$F_x = 1 - e^{-A/2\pi\sigma^2(R)}$ represents the probability of hitting the target area with a single round as a function of range, where

$\sigma (R^2) = a + bR + cR^2$

F_x effectively reduces the number of rounds hitting the target

These functions are derived in the Ground Fire Module write-up (Section 5).

Air-Ground Weapon Effects Function

One air-ground effects function is defined in the model, as follows:

Function 7 - Air Defense (used against aircraft)

$$E_k = 1 - (1 - P_H * P_{k/H}/NACFT) ** FIRE$$

where P_H = probability of a single round bit

$P_{k/H}$ = probability of kill given a hit

NACFT = number of aircraft as targets

FIRE = number of air defense shells fired

c) Term References:

Weapons effects function number is referenced by the Weapons Effects Table, and is a function of weapon type and target element type.

3.160.2 Programmatical Definition

a) Term Name:

IFEF

b) Term Description:

IFEF is a model generated variable that is used in subroutine FINFUN. It takes on a value of 2,3,5,6, or 7 depending on the Weapons Effects table match. The IFEF value corresponds with the function number discussed above. IFEF is set for each weapon of a single unit firing at a single target element in an enemy unit. IFEF is formally defined as follows: The weapon effects function number (1 through 7) to be used by a particular weapon against a particular target equipment or personnel vulnerability class.

c) Term References:

Subroutine FINFUN determines the proper setting of IFEF from the entry in table IFNTB, and subroutine EFFNS computes the appropriate function.

3.161 WEAPON-TARGET SET

3.161.1 English-language Definition

a) Term Name:

Weapon-target Set

b) Term Description:

Weapon-target sets are formed for the allocation of each direct and indirect fire weapon, first for Red units as firers, then Blue. For a given weapon, a unit on one side (say, Blue) containing the weapon is placed in the weapon set and examined for possible targets of the opposing color. All eligible target units for the weapon are placed in the target set.

The criteria to be met for a target unit to be eligible are:

1) Target unit is within range defined for firing unit "type"
2) For direct fire weapons (as opposed to indirect) no friendly unit blocks the line connecting the center of the target unit with the center of the firing unit.
3) Detection of the target unit is achieved by the firing unit
4) If both firing and target units are in the same engagement, and the weapon being considered is direct fire, they must both still be in the front line of an engagement.

For each of the target units placed in the target set, all opposing units (Blue in this example) are examined to determine if they meet the criteria to fire the weapon at the target unit. Any such fire unit is also placed in the weapon set, and target units not yet included in the target set are re-examined. When no additional units can be added to either set, the weapon-target set is complete. Additional weapon-target sets can be created for units on both sides not yet considered.

For each weapon-target set formed, a weapon-target subset is formed for each weapon to be fired (by Blue in this case) representing the which units will fire that specific weapon at which targets. The following criteria must be satisfied in forming the subsets:

1) Target unit is within primary range and has primary target elements of the weapon, or target unit is within secondary range and has secondary target elements of the weapon and no target units satisfy the primary range requirement.
2) Ammo is available in the unit for the weapon being fired.

Each unit in the weapon subset also has an individual target list of oppressing units eligible for that firing unit, with associated ranges.

c) Term References:

The use of weapon-target sets is completely restricted to the direct (and indirect) fire logic of the Ground Fire Module.

3.161.2 Programmatical Definition

a) Term Name:

IRTET(JWTS,J)
IT(JWTS)
ITET(JWTS,J)
IWT(JWTS)
LTGT(JWTS)
LWPN(JWTS)

b) Term Description:

The following arrays are model generated variables used, by the Ground Fire Module, in the formation of weapon-target sets, as described.

IRTET (JWTS,J)	For each entry in weapon-unit set JWTS, a list containing distance to Jth target unit at which it can fire (corresponds to ITET). Sometimes used as temporary storage.
IT (JWTS)	List of units in current target-unit set JWTS
ITET (JWTS,J)	Target eligibility table. For each entry in weapon-unit set JWTS, a list containing all Jth eligible target units.

IW (JWTS)	List of units in current weapons set JWTS.
IWT (JWTS)	Target allocation vector for unit JWTS. In a given weapon-target set, if unit is a weapon unit, IWT contains the number of enemy units at which it can fire. If unit is a target unit, IWT contains the number of enemy units which can fire at it (sometimes used as temporary storage).
LTGT (JWTS)	The JWTSth target subset (actual unit number).
LWPN (JWTS)	The JWTSth weapon subset (reflects the index of the weapon unit in LW)

Arrays IW and IT, combined, represent the weapon-target set. Arrays LWPN and LTGT, combined, represent the weapon-target subset. Arrays ITET and IRTET are lists of targets and their associated ranges for each entry in the weapon-target subset. Array IWTS is used for temporary storage in preparing the weapon-target set.

c) Term References:

Arrays IT, IW, LIGT, and LWPN can have up to 30 entries, each entry being a unit number. Arrays ITET and IRTET are dimensioned 30 x 8, allowing for up[redacted] Array[redacted]S is indexed by unit, IU (1 $\leq$ IU $\leq$ 100).

Subroutine WTSETS makes entries into arrays IW and IT, which are then used by subroutine WTSUB to determine entries into LWPN and LTGT, and also ITET and IRTET. These last four arrays are used by subroutine FIRALO to determine direct and indirect fire weapon allocations for all units.

3.162 WEATHER CLASS

3.162.1 English-language Definition

a) Term Name:

Weather Class

b) Term Description:

Weather class is modeled by a table of visibility and illumination data. Eleven classes of weather are distinguished. They are described as follows:

1) very clear
2) clear
3) light overcast
4) heavy overcast
5) haze
6) light rain
7) moderate rain
8) heavy rain
9) light fog or dust
10) moderate fog or dust
11) heavy fog or dust

Classes 6, 7, and 8 are modeled by the same data as classes 9, 10, and 11 respectively, thus the data table contains entries for eight weather classes rather than eleven.

Each weather class is represented by a set of eight data values. The first data value specifies the meteorological visibility of the weather class. The remaining seven data values specify various levels of light level throughout the day:

1) Day
2) Sunrise
3) Sunset
4) Night - no moon
5) Night - 1/4 moon
6) Night - 1/2 moon
7) Night - full moon

The data taken from this table along with general weather information taken from other tables determine the weather conditions in the area of operation.

The data table representing weather class is user defined by input; the table values remain constant throughout the simulation.

c) Term References:

Weather class is identified by the class number, an integer ranging from 1 to 8 inclusively. Note that the impact exerted by weather in the CATTS math model deals principally with detection. That is why weather class is characterized and distinguished by light and visibility data. The target acquisition (i.e., detection) module and the air module reference results computed from weather class data. The movement logic uses weather data to degrade movement during rain.

3.162.2 Programmatical Definition

a) Term Name:

VISLUM(8,8)

b) Term Description:

The floating point array VISLUM is a data base input variable that is defined in subroutine FORMAIN (data statement). It contains data representing eight different classes of weather. The CATTS math model, however, presents 11 classes of weather. The data describing the last three classes is exactly the same as the data describing the previous three classes, thus to conserve storage, data for eight classes is given. VISLUM is an 8 x 8 array such that each row defines a weather class and each column reference visibility and illumination data for that class.

VISLUM (IWC,J) Visibility and illumination data for weather class IWC

J=1 Meteorological visibility (meters)
2 Ambient light level (ft-lam.)-daylignt
3 Ambient light level (ft-lam.)-sunrise

4 Ambient light level (ft-lam.)-sunset
5 Ambient light level (ft-lam.)-night-no moon
6 Ambient light level (ft-lam.)-night-1/4 moon
7 Ambient light level (ft-lam.)-night-1/2 moon
8 Ambient light level (ft-lam.)-night-full moon

c) Term References:

The array VISLUM(IWC,J) is indexed by weather class number IWC ($1 \leq IWC \leq 8$). For each weather class, the index J ($1 \leq J \leq 8$) references the meteorological visibility (J=1) data, and the ambient light level (J=2 through 8) data. The array VISLUM is used in the following subroutines:

FORMAIN
INPUT
WETHR
WETHRC

3.163 XY POINT TARGET

3.163.1 English-language Definition

a) Term Name:

XY Point Target

b) Term Description:

When a fire control command is made, one of the choices for target type on the fire control menu is "XY point target". This option is only valid for support fire weapons. When this option is selected, an indicator is stored in the fire control table's listed of targets, and the point being fired at is stored. When the support fire weapon processing is performed, if a fire command is scheduled for a support fire weapon against an XY point target, the point is evaluated to determine if it falls within any unit's (friendly or enemy) area. If it does, weapons effects are computed against the unit as though the unit's center of mass was the aim point, consistent with other support fire effects evaluation is the model.

c) Term References:

XY point target is a specific designation for a support fire weapon and is set at the time the fire command is created, and used when the weapon is actually fired.

3.163.2 Programmatical Definition

a) Term Name:

II = 0

b) Term Description:

II is an interactively generated variable that is created from the fire menu. When II = 0, where II is used as the target unit number, special code in SPTALO is executed to call subroutine POINTGT and search over all units to determine if any were affected by the firing.

c) Term References:

Subroutine SPTALO checks for XY point target" indicators, and calls POINTGT to search for affected units.

4. DESCRIPTION OF DATA TABLES

CATTS is largely a table-driven simulator. Many of the tactical decisions and outcomes are calculated based on values found in input data tables. Some of these tables have complex structures, and there are often interactions between entities in several different tables. A knowledge and understanding of the structure, operation, and interaction of these tables is essential to the effective use of the CATTS system.

These tables, as are all the other arrays in CATTS, are filled with the numbers from the data base decks (weapon effects input deck, change of state input deck, suppression deck, etc.) and the other data disk files (DA, VEGCOMP, and DA, VEGLOC, specifically). A detailed discussion of the format of the data input to the tables discussed in this section, along with all the other scalar variables and arrays, are contained in the Data Base/Operations Manual. Additionally, definitions of the arrays comprising these tables are contained in Appendix A of this manual, Volume VII of the Programming Report, and Table 3-1 of the Data Base/Operations Manual. Thus, the purpose of this section is to explain, in detail, the workings of the most important and complicated of the CATTS data tables.

Section 4 will provide both English-language and programmatical descriptions of 27 major CATTS data tables. Also included will be any size limitations on the tables and references to the subroutines which access and/or modify each table. Also provided is Table 4-1 showing which deck of the data base or disk file contains the data to fill the tables and which subroutine reads in the data and packs it (if the data is packed). If the mathematical model creates the data in the tables or changes the data in the tables, this is clearly indicated, and the subroutines that create or change the data are listed.

Examples will be given wherever appropriate in order to clearly understand the actual and potential uses of each table. Frequent references will be made to specific terms in Section 3 that are related to a table definition. References will also be made to other tables in this section where a discussion of one table is not complete without a sufficient understanding of another.

Table 4-1. CATTS Table Data Origin and Use

Table Name	Where Data is Located or Created	Routines that Read, Pack, or Create Data	Other Routines that Access Data
Error Dispersion	Weapon Effects deck	Subroutine WEFINP	RDSON
Weapon Effects Coefficients	Weapon Effects deck	Subroutine WEFINP	DMGEVAL, EFFNS
Control Measure	Control Measure deck	Subroutine CMINP	ACTIVATE, CK4XING, CMSEGMNT, CNTRLMSF, CONTMS, FORRAM, LEAVEOG, NEWFWDUN, RMVOPGP
Movement Degradation	Math model created	Subroutine ADJROM	AMOVUL, ORGFIR
Engagement Expansion-Contraction	Movement deck	Subroutine MOVINP	OGFRNT
Ground Detection	Math model created	Subroutine DETECT	DTECTPRB, ENCTR, FIRALO, FORMAIN, OVERUN, STATREP, TGTLST, WPNEFF
Fire Control	Math model created	Subroutine FIRSORT	FIRALO, ISFIROR, SPTALO, SPTFIR, STATREP, STATREP1, SETIMP1
Weapon Effects Look-up	Weapon Effects deck	Subroutine WEFINP	FINFUN, SPTVSP, WPNEFF, WPNFIR
Unit Type Maximum Range	Unit Type deck	Subroutine UTINP	FIRELG
Fire Support	Fire Support deck	Subroutine FSPINP	SPTALO
Impacting Fires	Math model created	Subroutines ADW, AIRMOV2, SETIMP1, SETIMP2, and SPTFIR. Also main program CMAIN.	FORRAM
Change of State	Change of State deck. Model can delete one time entry.	Subroutine COSINP Subroutine CHGOPN	CHGCRT

Table 4-1. CATTS Table Data Origin and Use (Cont'd)

Table Name	Where Data is located or Created	Routines that Read, Pack, or Create Data	Other Routines that Access Data
Suppression Criterion Selection	Suppression deck	Subroutine SIGINP	SUPRES
Relief-Vegetation Association	Disk File VEGLOC on DA area of disk.	Subroutine LOSINP	LOSCOMP
Unit Vegetation and Soil Classes	Math model created.	Subroutines LOSCOMP and SOIL.	ADJROM, LOSINP, DLOSINP, INPUT, RADAR, UASCK, VISUAL
Minefield Data	Math model created from data read from Mine *Obstacle Forti*fication deck.	Subroutine OBFMINP reads the data base and sub*routine MINEFLDS creates* the table.	BRCHPATH, OBSTACLE, OBSWIDTH
Break Range	Equipment Mode deck	Subroutine MUINP	SCHMU
Mode Selection Code	Equipment Mode deck	Subroutine MUINP	SCHRMU
Movement Code Change	Movement deck	Subroutine MOVINP	NEWMOV
Movement Data for Operational Groupings	Operational Grouping deck	Subroutine OGINP	ADJDIR, ARRIVE, CHGCRT, CHGOPN, DEPLOC, DIRMOV, FWDLIN, LEAVEOG, MOVMNT, NEWENG, NEWFWDUN, NEWMOV, OGDIR, OGLOC, OGLOC2, OG2UNI, OPPLAN, PREMOV, RMVOPGP, STATREP, TASKORG, UNI2OG, UN2FEB, WTHDRW, ANYFOE

Table 4-1. CATTS Table Data Origin and Use (Cont'd)

Table Name	Where Data is Located or Created	Routines that Read, Pack, or Create Data	Other Routines that Access Data
Movement Data for Units	Unit deck	Subroutine UNINP	ADJDIR, ADJROM, AIREVENT, AIRMOV, ANYFOE, AREA, ARRIVE, CHGCRT, CHGOPN, DEPLOC, DIRMOV, ENCTR, FINDFO, FIRALO, FIRELG, FRNTGS, FWDLIN, HUMAN, INPUT, LATDST, MANEUVER, MOVE, MOVMNT, NEWENG, NEWFWDUN, NEWMOV, OBSDELAY, OBSUPDAT, OGDIR, OGLOC, OG2UNI, OPPLAN, OUTNAMLT, OVERUN, REL2FWDU, ROADCHK, SAVE, SAVEINP, SAVEOLD, STATREP, STEP, STLKUP, TASKORG, UNBLOK, UNI2OG, UN2FEB, USEFUEL, WTHDRW, OGINP
Vegetation Class data	Disk File VEGLOC on DA area of disk.	Subroutine LOSINP	LOSVEG
Suppression Curve	Suppression deck	Subroutine SIGINP	SUPRES
Soil Class Data	UGS deck	Subroutine SENSINP	ADJROM, FORMAIN, LOSINP, UASCK, VISUAL
Casualty Statistics	Math model created	Subroutines ADW, INIT, OBSDALAY, REDIST, STEP, and WPNFIR.	CASREP

Table 4-1. CATTS Table Data Origin and Use (Cont'd)

Table Name	Where Data is Located or Created	Routines that Read, Pack, or Create Data	Other Routines that Access Data
Mode Distribution Vector	Equipment Mode deck	Subroutine MUINP	ORGFIR, SPTALO, TGTCAT
Vegetation Polygon	Disk File VEGLOC on DA area of deck.	Subroutine LOSINP	LINEOBS, MICSOL

4.1 ERROR DISPERSION TABLE

The Error Dispersion Table consists of the following arrays:

IDISPR (10)

DISPER (10,3)

4.1.1 English-language Description

The dispersion table is used to estimate the number of shells falling within an enemy target unit for support fire weapons using weapons effects function 6. The table can have up to ten entries, one for each support fire weapon using effects function 6. The weapon numbers are listed in one array, and three associated coefficients are listed in three other arrays. The three coefficients, C_1, C_2, C_3, are parameters in the following equation:

$$\sigma^2(R) = C_1 + C_2R + C_3R^2$$

This equation is part of a Rayleigh distribution estimating the probability of a single round of the weapon falling within the target area, as a function of range, target area, and dispersion coefficients associated with the particular support fire weapon. The distribution is defined as

$$F\alpha = 1 - e^{-A/2\pi\alpha^2(R)}$$

where F represents the probability of hitting the target with a single round.

A represents the area of the target unit.

σ represents the standard deviation of the distance between impact point and unit center.

R represents the range between firing unit and target unit.

The distribution is deduced as follows:

Let x and y be two jointly distributed random variables, where x is the dispersion error in the vertical direction and y is the dispersion error in the horizontal direction.

Assuming that 1) x and y are uncorrelated and 2) $\sigma_x = \sigma_y = \sigma$ (variance of x and y are equal) the probability density function for

errors in x and y becomes:

$$f(x,y) = \exp[-(x^2 + y^2)/2\sigma^2]/2\pi\sigma^2$$

By approximating the target area with a circle of radius S and converting to polar coordinates via the substitutions

$$\left.\begin{array}{l} S = x\cos\theta \\ S = y\sin\theta \end{array}\right\} \; [S^2 = x^2 + y^2],$$

the density function simplifies to

$$f(x,y) = \exp[-S^2/2\sigma^2]/2\pi\sigma^2$$

To compute the probability of a given round falling within the circle of radius S (approximately within the target area), the following integration is performed

$$f_x = \int_0^S \int_0^{2\pi} f(x,y)\; dS d\theta$$

with the result that

$$f_x = 1 - \exp(-S^2/2\sigma^2).$$

Since A (area of target unit) = πS^2, substitution for S yields

$$f_x = 1 - \exp(-A/2\pi\sigma^2).$$

Finally, σ^2 is defined in the data base as a quadratic function of R,

$$\sigma^2(R) = a + bR + cR^2,$$

where, as before, R is the range to the target unit.

Data for the coefficients C_1, C_2, and C_3 was computed using estimates from the Joint Munitions Effectiveness Manual, Surface-to-Surface Series, as the basic source. This source provided data for Range and Deflection Probable Error for specified ranges which were converted to Circular Probable Error by the following equation:

$$CEP = 0.833 * RPE + 0.912 * DPE$$

The CEP, in turn, was converted to the statistical variance, assuming a

normal distribution, by the following equation:

$$CEP = 1.17 * \sigma^2$$

$$\sigma^2 = 0.855 * CEP$$

Finally, multiple regression analysis was performed on the error variances at specified ranges, yielding the three coefficients of a second degree polynomial as a function of range, $\sigma^2 = C_1 + C_2R + C_3R^2$. These coefficients are read as input at model initialization and stored in the dispersion table for the appropriate weapon.

It should be noted that the original source data reflected range and deflection errors for various ranges, where the errors were in units of meters, and the ranges were in units of kilometers. Therefore, the equation $\sigma^2 = C_1 + C_2R + C_3R$ assumes range values in kilometers.

4.1.2 Programmatical Description

The dispersion table is composed of two arrays, IDISPR (10) and DISPER (10,3). Subroutine WEFINP reads input data provided at model initialization time, and stores it in the arrays. Weapon (equipment) numbers (integer values ranging from 1 to 80) are input and stored in IDISPR, and the corresponding coefficients (floating point numbers) for that weapon are input and stored in DISPER. The table is illustrated below:

IDISPR (I)	DISPER (I,1)	DISPER (I,2)	DISPER (I,3)
.	.	.	.
.	.	.	.
.	.	.	.
Weapon Number	C_1	C_2	C_3
.	.	.	.
.	.	.	.
.	.	.	.

Subroutine RDSON is the only user of this information. When a weapon in a single fire unit against a single target unit is processed by the Ground Fire Module, subroutine WPNFIR calls RDSON to search the IDISPR array for a match on the weapon number. If no match is found, RDSON simply returns to WPNFIR and the fraction of rounds falling on the target unit is set to 1 by default. If a match is found at IDISPR (I), I is used as the index into the DISPER table to extract the three coefficients DISPER (I,1), DISPER (I,2), and DISPER (I,3). These coefficients are used in the equation

$$\sigma^2(R) = C_1 + C_2R + C_3R^2,$$

which subsequently is used in the equation

$$F\alpha = 1 - e^{-A/2\pi\sigma^2(R)}$$

Subroutine RDSON returns this value for $F\alpha$ in the variable EFFRAC, which is used in subroutine EFFNS to compute the lethal effects of the weapon.

A typical entry in the error dispersion table is as follows:

IDISPR (9)	DISPER (9,1)	DISPER (9,2)	DISPER (9,3)
35	.55995 E+2	-.1637 E+1	.542 E+0

IDISPR (9) is the ninth entry of the IDISPR array, containing an entry indicating weapon number 35. DISPER (9,1), DISPER (9,2), and DISPER (9,3) contain coefficients C_1, C_2, and C_3, respectively, associated with weapon 35. Weapon 35 is a 1.55 mm howitzer using weapons effects function 6. This example was taken from the FEBA Gold scenario.

4.2 WEAPON EFFECTS COEFFICIENTS TABLE

The weapon effects coefficients table is made up of the variable NXK and the following array:

EFFDAT(315,4).

4.2.1 English-language Description

The weapon effects coefficients table is a set of parameters used selectively for various weapons effects functions within the mathematical model. Seven weapon effects functions make use of this table. These functions calculate all damage perpetrated by ground units. The result of the function calculations represent expected casualties for the various target elements being considered. These casualties are accumulated until the end of the time step when all firing is completed so as not to reduce the opposing unit's firepower before it has had the opportunity to fire itself. Intermediate results are accounted for separately, however, so that personnel and equipment casualties resulting from one weapon type are not available as targets for a subsequent weapon type, since they have already been destroyed.

One of seven functions is used to calculate expected casualties (E_k) for ground fire. All seven functions are performed in subroutine EFFNS. The seven functions are defined below:

Function 1: $E_k = V*P_{k/_H}*[1 - \exp(-k/R^2)]$

Function 2: $E_k = V*P_k$

Function 3: $E_k = V*N*L/A$

Function 4: $E_k = T*N*C_{L/_{C_u}}$

Function 5: $E_k = N*[1 - \exp(-V*L/A)]$

Function 6: $E_k = N*[1 - \exp(V*L*F_x/A)]$

Function 7: $E_k = N*[1 - (1 - P_{k/_H}*P_{H/N})^V]$

where:

V = Volume of fire in rounds

$P_{k/H}$ = The probability of a kill, given a hit for a single round

$$K = \frac{(\text{presented area of target element in sq meters} * 10^6}{(\text{variance of aiming and dispersion error in mils}) * 2\pi}$$

R = Range between fire and target units

P_k = Probability of kill for a single round

N = Number of target elements

L = Lethal area

A = Target unit area fired into = length * width * fraction fired into

T = Expected number of target weapons lost for each crew lost in target unit

C_L = Number of personnel killed in target unit

C_u = Number of personnel in target unit

F_x = Fraction of rounds falling into area fired at.

The above parameters are determined as follows.

V is determined by the number of pieces of the weapon manned and unsuppressed multiplied by the rate of fire of the equipment.

$P_{k/H}$ is taken directly out of the table when used.

$\underline{K}$ is taken directly out of the table when used.

$\underline{R}$ is calculated as the distance between center of mass of target unit and center of mass of firing unit.

$\underline{P_k}$ is calculated as a quadratic function of range. The parameters of the quadratic function used are taken from the table.

$\underline{N}$ is the number of personnel in a particular vulnerability class or the number of pieces of a particular equipment type.

$\underline{L}$ is the lethal area of a round for effects functions 3, 5, and 6, as a quadratic function of range to the target. The parameters of the quadratic function used are taken from the table.

A is the unit area of the target fired at. It is computed from the unit area (length * width), reduced by the fraction of the unit fired at. Each unit has the fraction defined as part of its unit input definition. It allows for the concentration of unit elements in a subarea within the total area of the unit.

T is taken directly out of the table when used.

C_L is calculated for a single time step for an input specified vulnerability class. This vulnerability class is input via the table. If a zero is specified as the vulnerability class in question, then all personnel killed within the target unit are considered equally.

C_u is calculated for a single time step for an input specified vulnerability class. This vulnerability class is input via the table. As with C_L, a value of zero indicates all personnel within the target unit are to be considered.

F_x is the expected numbers of rounds falling inside the approximate target unit area. The derivation of F_x is included within the explanation of function 6.

The seven functions are derived below:

a) Function 1 - Aimed Fire (Used Against Personnel and Equipment

$$E_k = V*P_{k/H}*[1 - \exp(-k/_{R^2})]$$

This equation is derived as follows:

Let x and y be 2 jointly distributed random variables, where x is the dispersion error in the vertical direction and y is the dispersion error in the horizontal direction. Assuming that x and y are uncorrelated and that their respective variances are equal, the probability density function for errors in x and y becomes:

$$f(x,y) = \frac{1}{2\pi\sigma_m^2} \exp\left[-\frac{(x^2+y^2)10^6}{2\sigma_m^2}\right]$$

where σ_m^2 is the variance of the equal x and y dispersion errors in units. To compute the probability of a given round falling withing the circle of radius R (the probability of a hit), the following integration is performed:

$$\int_{|x| \leq R} \int_{|y| \leq R} \frac{1}{2\pi\sigma_m^{\;2}} \exp\left[-\frac{(x^2 + y^2)10^6}{2\,\sigma_m^{\;2}}\right] dy\, dx$$

The usual polar coordinates substitution reduces this to the following:

$$\int_0^R \int_0^{2\pi} \frac{1}{2\pi\sigma_m^{\;2}} \exp -\left[\frac{r^2\; 10^6}{2\,\sigma_m^{\;2}}\right] r\, dr\, d\theta = 1 - e\left(\frac{-R^2\; 10^6}{2\;{}_m^{\;2}}\right)$$

By substituting A_T for πR^2 (the presented target area), the probability of a hit becomes:

$$P_k = 1 - e\left(\frac{-\,A_T\; 10^6}{2\;{}_m^{\;2}}\right)$$

By substituting the k from above into this equation:

$$P_k = 1 - \exp\left[-\frac{k}{R^2}\right]$$

Finally, the expected number of kills can be shown to be

$$E_k = V{*}P_{k/_H}{*}P_H = V{*}P_{k/_H}{*}\left[1 - \exp\left(-\frac{k}{R^2}\right)\right]$$

the desired result.

b) <u>Function 2 - Aimed Fire (Used Against Personnel and Equipment)</u>

$$E_k = V{*}P_k$$

or:

E_k = (number of rounds fired) * (probability of a kill for a single round).

The following assumptions must be made in using this function for aimed fire purposes:

1) The targets are small in relationship to the area within which they lie.
2) No target is struck twice.
3) No target is aimed at by more than one weapon.

c) Function 3 - Area Fire (Used Against Personnel and Equipment)

$$E_k = VNL/A$$

$$= \frac{(\#\text{ rounds}) * (\#\text{ target elements in unit}) * (\text{lethal area of a round})}{(\text{area of target being fired at})}$$

The following assumptions must be made in using the function:

1) All target elements are in area fired into.
2) All target elements are distributed uniformly.
3) Lethal area of rounds fired do not overlap.
4) All rounds land within the target area.

Here, $L/_A$ is used to represent the fraction of the target area that is made lethal to the target equipment by a single round. Thus, VL/A of the target equipment is destroyed by V rounds. N multiplied times this number gives the actual number of targets destroyed by V rounds.

d) Function 4 - (Equipment Targets Only)

$$E_k = T*N*C_{L/C_u}$$

This function allows equipment losses to be made proportional to personnel losses of personnel operating the target equipment. The validity of this function can be seen by considering the following:

C_{L/C_u} is simply the fraction of designated personnel within a given time step who have been killed within the target unit. The personnel considered are either all of those within a unit or only those who belong to a specified vulnerability class.

$N*(C_{L/C_u})$ is thus the number of the target crews lost if the target crew losses are assumed to be in the same ratio as the total personnel losses.

$T*N*(C_{L/C_u})$ is then simply the number of equipment losses. T need not necessarily have the value of 1. Non-armored equipment may be much less vulnerable to certain kinds of enemy fire than the

crew which operates it. Thus, T may be thought of as the probability of equipment loss given that the operating crew has been killed.

e) Function 5 - Area Fire (Used Against Personnel or Equipment)

$$E_k = N * [1 - e(-VL/A)]$$

This equation is derived as follows:

1) Probability of being within lethal area of a single round = L/A
2) Probability of not being within the lethal area of that round = 1 -L/A
3) Probability of not being within the lethal area of V rounds = $(1 - L/A)^V$
4) Probability of being within the lethal area of one or more rounds = $1 - (1 - L/A)^V$
5) For V large and (L/A) small, $(1-L/A)^V$ can be approximated by e^{-VL}/A by the Poisson Theorem
6) Therefore $1 - e^{-VL/A}$ represents the probability of each of the N target elements being in the lethal area
7) Since the N target elements are distributed uniformly, $E_k = N* \left[1-e^{-VL/A}\right]$, the number of casualties expected.

Function 5 differs from function 3 only be accounting for the overlapping of lethal areas produced by the rounds. It does not account for the possibility of rounds landing outside of the target area.

f) Function 6 - Area Fire (Used Against Personnel and Equipment)

$$E_k = N*[1 - \exp(-VLF_x/A)]$$

This equation is the same as function 5 except that $F_x = 1 - e^{-A/2\pi\sigma^2(R)}$ represents the probability of hitting the target area with a single round as a function of range, where $\sigma^2(R)$ is the standard deviation of the aiming error of a single round as a function of range.

This is a Rayleight distribution describing the probability of each round falling within the target area.

This distribution is deduced as follows:

Let x and y be two jointly distributed random variables, where x is the dispersion error in the vertical direction and y is the dispersion error in the horizontal direction.

Assuming that 1) x and y are uncorrelated and 2) $\sigma_x = \sigma_y = \sigma$ (variance of x and y are equal), the probability density function for errors in x and y becomes:

$$f(x,y) = \exp[-(x^2 + y^2)/2\sigma^2] / 2\pi\sigma^2.$$

By approximating the target area with a circle of radius S and converting to polar coordinates via the substitutions,

$$\left.\begin{matrix} S = x \cos\theta \\ S = y \cos\theta \end{matrix}\right\} \left[S^2 = x^2 + y^2\right]$$

the density function simplifies to:

$$f(x,y) = \exp[-S^2/2\ ^2]/2\ \ ^2.$$

To compute the probability of a given round falling within the circle of radius S (approximately within the target area), the following integration is performed:

$$F_x = \int_0^S \int_0^{2\pi} f(x,y)\, dSde$$

with the result that

$$F_x = 1 - \exp(-S^2/2\sigma^2).$$

Since A (area of target unit) = πS^2, substitution for S yields:

$$F_x = 1 - \exp(-A/2\pi\sigma^2).$$

g) Function 7 - Aimed Fire (Used Against Airborne Equipment)

$$E_k = N * [1 - (1 - P_{k/_H} * P_H/N)^V]$$

This equation is derived as follows:

1) Assuming a random target-weapon assignment, each weapon round has a probability of 1/N of attacking any given target.
2) The probability that the i$^{\underline{th}}$ target will be killed by the first weapon round is P_k/N.
3) The probability that it will survive the first weapon round is $1 - P_k/N$.
4) The probability that it will survive all weapon rounds is $(1 - P_k/N)^V$.

5) Thus, the probability that the $i^{\underline{th}}$ target will be killed is $1 - (1 - P_{k/N})^V$.

6) Since $P_k = P_{k/H} * P_H$, this becomes $1 - (1 - P_{k/H} * P_{H/N})^V$.

7) For all targets, then, the number of expected kills is
$$E_k = N * [1 - (1 - P_{k/_H} * P_{H/N})^V].$$

Function 7 assumes a random assignment of weapons and rounds to possible targets.

4.2.2 Programmatical Description

The weapon effects coefficients table is composed of one array EFFDAT(315,4). Subroutine WEFINP reads input data provided at model initialization and stores it in the array. The purpose of this table is to store the coefficients used by the seven weapon effects functions. Subroutine WPNFIR accesses this table through pointers found in the weapon effects look-up table (IFNTB). For every weapon-target combination exercised, the appropriate entry in IFNTB is found and used to determine the correct function to utilize and the location of the parameters of that function in EFFDAT.

The following sections define the coefficients contained in EFFDAT for each of the seven functions. In the discussion, the parameters K_i are used to designate table entry EFFDAT(I,i).

a) Function 1

$$E_k = Vk_1 [1 - \exp (-k_2/D^2)]$$

where

k_1 = probability of kill given a hit

$D = \text{Sup} \{k_3, R\}$.

K_3 is, in effect, the minimum range to be used in Function 1. This function is commonly used to estimate the effect of aimed fire subject to normally distributed aimining and dispersion errors. When it is so used (with R and k_2 expressed in meters),

$$k_2 \equiv \frac{(\text{presented area of target element in square meters}) * 10^6}{(\text{variance of aiming and dispersion error in mils}) * 2\pi} .$$

Note that k_4 in the grouping of four input constants is not used with this function and should be set to zero.

b) Function 2

$$E_k = V(a + bR + cR^2)$$

where

$a = k_1$, $b = k_2$, $c = k_3$ when $0 \leq R < k_4$

$a = k_5$, $b = k_6$, $c = k_7$ when $k_4 \leq R < k_8$.

When any of the constants k_4, k_8, k_{12}, . . . , is set to zero, the last set of coefficients given is to be used for all greater values of R.

This function permits the piece-wise fitting of any empirical weapon-effects curve by a set of quadratic functions. Each quadratic is defined by a grouping of four input constants, where the last constant in the last grouping to be used is set to zero.

c) Function 3

$$E_k = VNL/A$$

where

A = (width X depth of target unit) x (fraction of target unit actually fired at),

$L = (a + bR + cR^2)$

and

$a = k_1$, $b = k_2$, $c = k_3$ when $0 \leq R < k_4$

$a = k_5$, $b = k_6$, $c = k_7$ when $k_4 \leq R < k_8$.

When any of the constants k_4, k_8, k_{12}, . . ., is set to zero, the last set of coefficients given is to be used for all greater values of R. L represents a "lethal area," and if its value varies with range, it may be approximated piece-wise by quadratic functions, using a procedure similar to that used for function 2.

d) Function 4

$$E_k = \frac{k_1 N \left[\begin{array}{l}\text{Unit personnel losses during time step due to effects} \\ \text{of given weapon type against vulnerability class } k_2\end{array}\right]}{[\text{No. of target unit personnel in vulnerability class } k_2]}$$

$$= k_1 N \left[\frac{[\text{No. of target weapon crews lost}]}{[\text{No. of target weapon crews in target unit}]}\right]$$

where

k_1 = expected number of target weapons lost to fire for each weapon crew lost. This kind of factor can also be used with nonweapons that are target elements.

k_2 = personnel vulnerability class of crews operating the target equipment. $k_2 = 0$ implies that more than one personnel vulnerability class is involved, and consequently, E_k is computed, using the total numbers of personnel and personnel losses in the target unit.

This function permits the user to make equipment losses proportional to personnel losses (for personnel operating the target equipment). k_3 and k_4 in the grouping of four input constants are not used with this function and should be set to zero.

e) Function 5

$$E_k = N[1 - \exp(-VL/A)]$$

where L and A are defined exactly as they are in function 3. Function 5 provides a better estimate of E_k than does function 3, when lethal areas become large enough compared to the area fired into, that overkill becomes a significant factor.

f) Function 6

$$E_k = N * [1 - \exp(-V * L * F_x/A)]$$

This function is essentially the same as Function 5, except that the rate at which rounds fall on the target unit is reduced to reflect the fact that some rounds will fall outside the unit's area. This reduction in the volume of fire on the target is determined by a dispersion factor, F_x, which is computed as a function of range by use of the Error Dispersion Table.

g) <u>Function 7</u>

$$E_k = N * [1 - (1 - k_2 * k_1/N)^V]$$

where

$k_1 = P_H$ = *probability of a hit*

$k_2 = P_{k/H}$ = probability of a kill given a hit.

4.3 CONTROL MEASURE TABLE

The Control Measure Table consists of the following array:

ICM (15,100)

4.3.1 English-language Description

The control measure table is used to store information required for describing all control measures that are currently active in the CATTS mathematical model. A given control measure is described by a set of 6 atrributes. The attributes have the following meanings:

1. Control measure type (boundary line, objective area, etc.)
2. Unit to which the control measure applies. The control measure affects this unit and all units subordinate to this unit.
3. The red or blue affiliation of the control measure. Control measures are considered to be pertinent to red commanders, blue commanders of both.
4. The model applicability of the control measure. Some control measures are used for modeling purposes and others are for visual display only. Those that are used for modeling purposes can be used to check for unit crossings, fire crossings or firing short of the control measure. Alerts are generated indicating which of the above violations has occurred.
5. The location of the control measure.
6. The alphanumeric name of the control measure.

The control measure table is first set up during model initialization. During this period, the control measures which are part of the standard scenario being run are input to the table. During the simulation, control measures are input to the table. During the simulation, control measures are added to and deleted from this table by interactivity by the controllers via the events processor. Control measure events are processed at the beginning of each time step by the control measure events processor. Requests for additions to the table are processed until the table becomes full. At this time, requests for more additions are ignored and RAMALERTS are sent to the requesting controllers indicating this fact.

The control measure table is accessed, but not modified, by the foreground routines, the firing routines, the ground movement routines, the air movement

routines and others. The foreground routines access the table in order to display their positions as part of the graphic display functions and to use their names for selection on the command and control menus. The firing routines access the table in order to check for firing over and short of control measures. The ground movement routines access the table to check for units crossing designating control measures and to find control measures defining roads which are to be followed. The air moveme.t routines access the table to determine whether prohibited areas or lines are being crossed. Several other routines access the table. These routines check for such things as table location in memory and control measures belonging to operational groups

4.3.2 Programmatical Description

The array ICM is used dynamically during any given simulation. A standard set of entries are input to ICM at the beginning of each simulation. These input control measures are part of the scenario data base. During a given simulation, controllers use the control measure command and control functions to add, delete and move control measures. The immediate effect of these actions is to add, delete and change the entries in array ICM. This is accomplished by searching through ICM for the entry in question for deletions and changes and by searching for unused entries in order to make additions. The table is organized as follows:

> ICM (I,J), where I is the entry number and J is the index in the given entry.
>
> Unused entries are indicated where ICM (I,1,) is zero. Entry I is deleted by setting ICM (I,1) to zero.

Each control measure has 6 attributes which completely describe it. Up to 15 words of memory are used to define the six attritutes. All 15 words are reserved for each control measure even though not all of them are necessarily used. Index J of ICM (I,J) is used to reference the Jth word of control measure I. The values associated with each J are defined as follows:

J=1, This word is zero if the control measure is not in use. Otherwise this word is used as a byte packed array with 4 attributes stored within it.

> byte 0 = Type number of this control measure.
> byte 1 = Unit number of unit to which the control measure applies.

A given unit and all of its subordinate units are controlled, aided or otherwise required to utilize a control measure to which they are assigned in this manner.

byte 2 = Flag indicating whether the control measure applies to both red and blue units (value 0), red units only (value 1), or blue units only (value 2).

byte 3 = Flag indicating what model interaction with this control measure is required. Some control measures, such as Axis of Advance, require no model interaction. They are used for display purposes only by the foreground software. Such non-model control measures have a value of zero in byte 3.

A value of N = 1 to 63 means that an alert of type N should be generated whenever a unit to which the control measure applies crosses the control measure.

A value of N = 64 to 127 means that an alert of type N - 64 should be generated whenever a unit to which the control measure applies fires support fire weapons across the control measure.

A value of N greater than 127 means that an alert of type N - 128 should be generated whenever a unit to which the control measure applies fires support fire weapons short of the control measure.

It should be noted that firing violation alerts are only generated when a controller interactively generates a fire command which causes one of the above conditions to be met.

J = 2, The minimum X portion of the 10 digit map coordinate of any point on the control measure.

J = 3, The maximum X portion of the 10 digit map coordinate of any point on the control measure.

J = 4, The minimum Y portion of the 10 digit map coordinate of any point on the control measure.

J = 5, The maximum Y portion of the 10 digit map coordinate of any point on the control measure.

J = 6 to J = 13, The X,Y coordinates of the points defining the control measure. Point J-5 is defined by word J where the left-hand half-word of J added to the minimum X value (word 2) and the right-hand half-word of J added to the minimum Y value (word 4) represent the X,Y coordinate of the point. If less than 8 X,Y coordinates are required to define the control measure, those coordinates

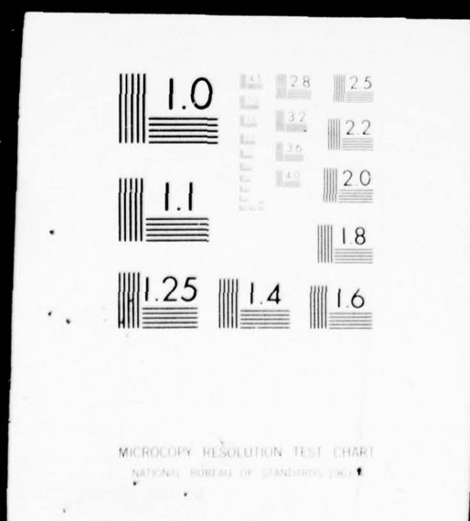
AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

7 OF 8
AD
A038796

7 OF 8

AD
A038796



(words ICM(I,J)) that are not used are set to -1. Control points are defined by small rectangles centered about the control points true location. In this case, the true location of the control point is given by X = ICM(11,I), Y = ICM(12,I).

J = 14 to J = 15 is the EBCDIC representation of the alphanumeric name of the control measure.

The control measure array is accessed by several routines within the math model. These routines and the purposes of the accesses are listed below:

ACTIVATE - When a unit is deactivated, all control measures assigned to it are deleted.

CK4XING - This utility routine is used for determining whether units cross control measures by air or ground and whether they fire support weapons across or short of specified control measures. Routines accessing the control measure table through CK4XING are listed below:

AIRMOV2 - Requests determination of whether air units cross control measures illegally.

OVERUN - Requests determination of whether ground units cross control measures illegally.

SETIMP1 - Requests determination of support fire violations.

SPTALO - Requests determination of support fire violations.

CMINP - The initial control measures are read in by this routine.

CMSEGMNT - This routine determines the next point on the "ROUTE" control measure for a ground unit to travel towards.

CONTMS - The control measure events created by the foreground command and control routines are passed to this routine for placement into the control measure table.

FORRAM - This routine passes a pointer to the beginning of the control measure table to the foreground routines.

LEAVEOG - Control measures that belong to operational groups that are disbanded due to too few units belonging to the operational group after a unit has left it are deleted from the control measure table.

NEWFWDUN - Control measures that belong to operational groups that are determined to be defunct are deleted from the control measure table.

RMVOPGP - Control measures that belong to operational groups that are disbanded due to too few units belonging to the operational group after a unit has been removed from it are deleted from the control measure table.

4.4 MOVEMENT DEGRADATION TABLE

The Movement Degradation Table is comprised of data found in the byte-packed FORTRAN array IDEGRAD (350).

4.4.1 English-language Description

Environmental features generally exert a retarding or degrading effect on equipment movement rate. The effect is quantified and used to determine overall unit movement rate. The technical approach taken to represent the effect involves deriving several degradation factors for each of several environmental features being modeled. This approach is based mainly on the Vehicle Speed Prediction Model developed by the Defense Mapping Agency. The underlying premise of this model is that the maximum operational speed of a vehicular equipment is attained on a firm, smooth flat surface under optimum conditions. As conditions depart from the optimum, the speed is diminished by an amount proportional to the degree of departure from the optimum. The diminishing effects are represented by a set of multiplicative factors, each factor being independent of the others.

The amount of movement degradation experienced by vehicular equipment is computed differently between onroad and offroad travel. Offroad travel is affected by the following considerations:

1. relief (terrain slope)
2. vegetation
3. soil
4. micro-relief
5. visibility

Onroad travel is generally less constraining because roads are usually free of obstructive terrain features. The following considerations have a significant influence in determining onroad movement rate:

1. relief (terrain slope)
2. visibility
3. type of road
4. damage of road due to air and ground ordinance

In addition to the above factors, movement degradation due to human fatigue is modeled whenever the unit is traveling in the dismounted mode.

Each environmental degradation factor is represented by a fraction ranging in value from zero to one inclusively. For every self-propelled equipment type within a unit, a set of fractions is derived and used as multiplicative factors

to determine one overall fraction. This single fraction will represent the combined effects of all environmental features which affect the mobility of the equipment type. The combined factor is applied to the maximum operational speed of that equipment type. This produces a degraded top speed attainable by the equipment type. The speed ultimately assigned is determined to be the slower rate between the degraded top speed mentioned above, and the average speed as computed by the ground fire logic.

Each unit may have as many as fourteen different types of equipment. A degradation fraction is computed for each equipment type per time step. Data used to derive each fraction exists in various data base tables established to model environment. The fraction is converted into an integer percent and stored into memory to be referenced later (during the same time step) when unit movement rates are to be calculated. Every equipment type within the unit is assigned a speed which depends upon the following:

1. the environmental degradation factor mentioned above
2. The distribution of each equipment over its modes of operation
3. the input-defined speeds of the equipment type operating in its various modes of operation
4. the amount of suppression experienced by the unit

Note that unit movement rate is determined by the slowest moving equipment type in that unit.

4.4.2 Programmatical Description

The Movement Degradation Table is represented by data stored in the byte-packed array IDEGRAD(350). Every equipment type in each unit has an environmental degradation factor computed per time step. Since a maximum of 100 units may exist, each of which may have a maximum of 14 different types of equipment, 1400 (100 x 14) bytes of memory must be reserved to establish the table (hence 350 words = 350 x 4 = 1400 bytes). Data in each byte exist in the form of an integer, representing percent of degradation. It is retrieved by indicating the unit number, and the position of the equipment type in the unit's equipment list. For example, to obtain the percent degradation for the J-th equipment in unit I's equipment list, reference the K-th byte from the first byte (byte 0) of array IDEGRAD, where K = (I-1)x 14 + J - 1. Note that J ranges in value from 1 to 14 because a unit is allowed at most 14 different types of equipment in its equipment list, and I ranges from 1 to 100 because a maximum 100 units may be modeled.

As the indexing described above implies, data is organized by unit according to the unit's equipment list. The array IDEGRAD can be thought of as a series of 1400 bytes, such that every multiple of 14 bytes (starting from byte 0) contains the percent degradations to be applied to the equipment in a unit. For instance, bytes 0 through 13 contain the degradation percents for the fourteen equipment types of unit 1, bytes 14 through 27 contain the percents for unit 2's equipment list, etc. Table 4-2 illustrates the structure and organization of the Movement Degradation Table.

Subroutine ADJROM, utilizing data from the terrain and weather tables, establishes a percent degradation for every self-propelled equipment type within a unit. This data is stored in common to be referenced by the ground fire subroutines (AMOVUL and ORGFIR) in calculating unit movement rates. Note that the array IDEGRAD is dynamic, meaning that the data contained within it is constantly being updated (every time step), depending upon changes in unit location and thus environmental surroundings.

Table 4-2. Array IDEGRAD Representing the Movement Degradation Table

Word 1	Byte 0 % degrad. of 1st equip. in unit 1	Byte 1 % degrad. of 2nd equip. in unit 1	Byte 2 % degrad. of 3rd equip. in unit 1	Byte 3 % degrad. of 4th equip. in unit 1
Word 2	Byte 4 % degrad. of 5th equip. in unit 1	Byte 5 % degrad. of 6th equip. in unit 1	Byte 6 % degrad. of 7th equip. in unit 1	Byte 7 % degrad. of 8th equip. in unit 1
Word 3	Byte 8 % degrad. of 9th equip. in unit 1	Byte 9 % degrad. of 10th equip. in unit 1	Byte 10 % degrad. of 11th equip. in unit 1	Byte 11 % degrad. of 12th equip. in unit 1
Word 4	Byte 12 % degrad. of 13th equip. in unit 1	Byte 13 % degrad. of 14th equip. in unit 1	Byte 14 % degrad. of 1st equip. in unit 2	Byte 15 % degrad. of 2nd equip. in unit 2
Word 5	Byte 16 % degrad. of 3rd equip. in unit 2	Byte 17 % degrad. of 4th equip. in unit 2	Byte 18 % degrad. of 5th equip. in unit 2	Byte 19 % degrad. of 6th equip. in unit 2
. . . .	. . . .	. . . .	. . . .	. . . .
Word 347	Byte 1384 % degrad. of 13th equip. in unit 99	Byte 1385 % degrad. of 14th equip. in unit 99	Byte 1386 % degrad. of 1st equip. in unit 100	Byte 1387 % degrad. of 2nd equip. in unit 100
Word 348	Byte 1388 % degrad. of 3rd equip. in unit 100	Byte 1389 % degrad. of 4th equip. in unit 100	Byte 1390 % degrad. of 5th equip. in unit 100	Byte 1391 % degrad. of 6th equip. in unit 100
Word 349	Byte 1392 % degrad. of 7th equip. in unit 100	Byte 1393 % degrad. of 8th equip. in unit 100	Byte 1394 % degrad. of 9th equip. in unit 100	Byte 1395 % degrad. of 10th equip. in unit 100
Word 350	Byte 1396 % degrad. of 11th equip. in unit 100	Byte 1397 % degrad. of 12th equip. in unit 100	Byte 1398 % degrad. of 13th equip. in unit 100	Byte 1399 % degrad. of 14th equip. in unit 100

4.5 ENGAGEMENT EXPANSION-CONTRACTION TABLE

The Engagement Expansion-Contraction Table is comprised of data entries residing in the array IEXTBL(15,2).

4.5.1 English-language Description

When two opposing operational groupings engage one another, there may be a requirement for one of them to adjust its frontage to fit that of the other. This is done by spreading the operational groupings units farther apart laterally or by moving them closer together. Similarly, if an operational grouping joins an existing engagement, it may choose to adjust its frontage to that of the friendly force already in the engagement. In either case, such an adjustment is accomplished by using the Engagement Expansion-Contraction Table.

Currently, a maximum of fifteen data entries make up the table. Each entry consists of a pair of computer words packed with two kinds of information:

1) Data stipulating conditions and characteristics which must be satisfied by the operational grouping before the expansion or contraction can be achieved.

2) Data pertaining to the expansion/contraction procedure.

This information is established by the user at his discretion through inputs. The table is initialized at model start time and remains constant throughout the simulation.

The operational grouping whose frontage may be modified is matched against data entries in the Engagement Expansion-Contraction Table. If an entry in this table describes (i.e., matches) the operational grouping, then the grouping's expansion-contraction factor is changed so that its deployed frontage will fit the frontage of either the enemy or friendly force, as stipulated by the entry. The "normal" lateral spacing of units in an operational grouping (i.e., the spacing defined by input) is multiplied by the expansion-contraction factor of the operational grouping to obtain the new expanded or contracted lateral unit spacing. When a new engagement between two operational groupings is formed,

the force that initiated the engagement retains its normal frontage, while the other force is given the opportunity to adjust to this frontage. On the other hand, an operational grouping that joins an already existing engagement is always given the option of adjusting its frontage either to that of the friendly or enemy force. These changes in engagement frontage are controlled by the user through decision rules stored in the Engagement Expansion-Contraction Table.

4.5.2 Programmatical Description

The Engagement Expansion-Contraction Table is comprised of data entries residing in the array IEXTBL(I,J). Currently a maximum of fifteen data entries, indexed by the array subscript I ($1 \leq I \leq 15$) may be defined for the table. Each entry consists of a pair of computer words, indexed respectively by J ($1 \leq J \leq 2$), packed with information used to modify the frontage of an operational grouping entering an engagement. The frontage is made to fit the friendly or enemy frontage according to user defined codes stored as follows:

IEXTBL (I, 1) = UUVV

IEXTBL (I, 2) = WXXYYZ

(1 < = I < = 15)

Such that:

UU = Operational grouping number (00 = > any number),

VV = Characteristic unit type of operational grouping (00 = > any type),

W = Red grouping (1) or blue grouping (2) (0 = > either),

XX = Characteristic unit type of controlling enemy operational grouping in engagement (00 = > any type),

YY = Characteristic unit type of controlling friendly operational grouping (00 = > any type),

Z = Expansion-Contraction code:
1 = > Adjust to fit enemy frontage
2 = > Adjust to fit friendly frontage.

IEXTBL is a static array: once its values have been established by input (subroutine MOVINP) during model initialization, they are not modified by subsequent processing.

The array IEXTBL is accessed only by subroutine OGFRNT. This subroutine calculates the frontage expansion-contraction factor (given by one FORTRAN variable FECFAC) whenever an operational grouping engages the enemy. OGFRNT is called with two arguments:

J = the operational grouping number which has become engaged

K = the number identifying the engagement it has just entered.

OGFRNT first checks to determine whether operational grouping J is non-defunct, and whether the K-th engagement contains both red and blue controllings operational groupings. If not, modifications are not required; OGFRNT returns immediately. On the other hand, if both of the above conditions are satisfied, the array IEXTBL is searched to decide whether to adjust the frontage of the friendly or enemy controlling operational grouping. The search examines the following factors:

1) The type of grouping of operational grouping J
2) The color (red or blue) of operational grouping J
3) The type of grouping of the enemy controlling operational grouping.

These are the primary factors which determine whether a match is found. If no match occurs, OGFRNT returns without modifying the frontage. Otherwise, the number of the appropriate controlling operational grouping (call this number JP) is unpacked from the matching code word. The integer JP is used to index into arrays FECFAC and HLFRN respectively. These arrays provide data necessary to modify the frontage. The modification is accomplished by the following FORTRAN expression:

```
FECFAC(J) = FECFAC(JP) * HLFRN(JP)/HLFRN(J)
```

where

J = operational grouping number which has become engaged

JR = number of the appropriate controlling operational grouping

FECFAC(J) = the frontage expansion-contraction factor for the J-th ($1 \leq J \leq 20$) operational grouping

HLFRN(J) = the normal half-frontage of the J-th ($1 \leq J \leq 20$) operational grouping.

The structure and organization of the array IEXTBL representing the Engagement Expansion-Contraction Table is illustrated below:

ENTRY NO.1	WORD 1	WORD 2
1	IEXTBL(1,1)	IEXTBL(1,2)
2	IEXTBL(2,1)	IEXTBL(2,2)
.	.	.
.	.	.
.	.	.
N	IEXTBL(N,1) UUVV	IEXTBL(N,2) WXXYYZ
.	.	.
.	.	.
.	.	.
15	IEXTBL(15,1)	IEXTBL(15,2)

4.6 GROUND DETECTION TABLE

The Ground Detection Table consists of the following arrays:

IFIREFA(100,25), IDETV(4,100).

4.6.1 English-language Description

One of the primary functions of the Target Acquisition Module is to determine which opposing ground units are detected each minute by each active ground unit. In addition, for each detection, the fraction of the opposing unit detected is computed. These results are stored in the arrays comprising the Ground Detection Table, the detection indicator stored in IDETV, the fraction in IFIREFA.

At model initialization time, the table is cleared with all zeros. Each time step during the exercise, the probability of non-concealment of a target unit with respect to an observer unit is computed by the line-of-sight function and stored for use by the ground detection function.

For each ground unit matched against each opposing ground unit of the enemy, the ground detection function checks the probability of non-concealment this time step and the detection indicator and the fraction of the unit detected last time step. These inputs are used to determine the detection indicator and fraction for the current time step. One of the following conditions is encountered:

1) If the fraction of non-concealment is zero (zero probability of detection this time step) and the detection fraction from last time step is zero, then both the detection indicator and fraction for this time step are set to zero.

2) If the fraction of non-concealment is zero and the detection fraction from last time step is greater than zero, then the detection indicator is set to zero and the detection fraction is reduced (aged) by a degradation factor.

3) If the fraction of non-concealment is greater than zero and the detection indicator from last time step is set, then the larger of the previous detection fraction and the fraction of non-concealment is stored and the detection indicator remains set this time step if detection does not occur.

4) If the fraction of non-concealment is greater than zero and the detection indicator from last time step is not set, then the fraction of non-concealment is stored and used to compare a random number against to determine if detection should occur this time step; if detection does not occur as a result of the random number draw, the detection fraction is set to zero.

Each new detection in a given time step is accompanied by the issuance of a visual detection alert identifying the observer, observee, and percentage detected.

The detection fraction between pairs of units, one the observer, the other the unit being observed, is used primarily by the ground fire function. Direct and indirect fire weapons weight each potential target by its detection fraction. If the detection fraction is zero, direct and indirect fire weapons will not allocate fire against a potential target unit.

Weapon effects for direct and indirect fire weapons are also reduced proportionately by the fraction of the target unit detected.

The engagement logic also utilizes the detection fraction to determine if an engagement could be formed. If the fraction is zero, an engagement between units cannot be formed.

4.6.2 Programmatical Description

Subroutine DETECT is the program that sets both arrays IDETV and IFIREFA at the beginning of each time step so that other functions executing later in the time step, namely the ground fire and engagement functions, can utilize the detection data. The principal subroutines using the data and their reason for using it are as follows:

FIRALO	Allocate direct and indirect fire weapons according to the fraction of target units detected.
WPNEFF	Reduce casualties for direct and indirect fire weapons as a function of the fraction of the unit detected (and, hence, fired into).
ENCTR	An engagement cannot be formed with a unit having a detection fraction of zero.

The array IFIREFA contains the fraction of the opposing unit detected by a ground unit. The array is dimensioned 100x25 and contains 10,000 bytes of information (100*25*4). The 10,000 bytes contain detection information for 100 units as observers, each of which can have up to 100 units as Target. Each byte contains a value ranging from 0 to 100, representing the fraction of target detected. Clearly only about one-half of this table is used, since red units do not record detection of red units, nor blue against blue. However, for ease of indexing and faster execution, a more complicated indexing scheme minimizing table storage requirements was not implemented.

The first index, I, of the array IFIREFA(I,J) is the unit number of the observer unit. The second index, J, is a function of the number of the observed unit, K, and is computed as follows:

$$J = (K - 1)/4 + 1$$

The correct byte, IBYTE, is determined by:

$$IBYTE = MOD(K - 1,4).$$

For instance, if blue unit 52 was detecting red unit 16, the result would be stored in the third (right-most) byte of IFIREFA(52,4), computed as:

$$J = (16-1)/4 + 1 = 3 + 1 = 4$$

$$IBYTE = MOD(16-1, 4) = MOD(15,4) = 3$$

The detection indicator array, IDETV, indicating whether or not detection actually took place in the present time step, is a bit-packed array dimensioned 4*100. IDETV(I,J) is indexed such that J is the observer unit number. The value of I is a function of the observed unit number, K, and is computed by:

$$I = (K + 31)/32.$$

The circular shift count, ISHIFT, for unit K within word I is determined as follows:

$$ISHIFT = MOD(K,32).$$

The actual bit position is ISHIFT-1 (if ISHIFT=0, bit position = 31), where the left-most bit is position 0. Using the above example of unit 52 observing unit 16, the result would be stored in the fifteenth bit of IDETV(1,52), and the value of ISHIFT would be 16.

The computation would be as follows:

$$I = (16 + 31)/32 = 1$$

$$ISHIFT = MOD(16,32) = 16$$

$$\text{Bit Position} = ISHIFT - 1 = 16 - 1 = 15.$$

4.7 FIRE CONTROL TABLE

The Fire Control Table consists of the following two arrays:

IFIROVRD (100)

ITGTLST (100,10)

4.7.1 English-Language Description

The Fire Control Table is the storage area in which all fire commands are saved. These commands are issued via the Fire Control menu, and consist of the following information:

- Unit being commanded to fire
- Specific weapon in the unit being commanded to fire
- Duration of the command
- List of targets against which the weapon is commanded to fire (up to a maximum of 8)
- Number of rounds, or percentage of the weapon's firepower, to be fired at each target
- For each area target, a pointer to the location in the Impacting Fires Table where the X, Y location is to be saved
- Type of fire command, basically consisting of one of the following:
  - --- Fire a specified percentage of firepower at each target designated
  - --- Fire a specified number of rounds at each target designated
  - --- Cease fire from the weapon for the specified duration
  - --- Cancel the existing fire command for the specified weapon

All this information along with the scheduled time of occurance is stored in an events file. When the scheduled time arrives, the data associated with the event is processed by the fire event function, which stores appropriate data in the Fire Control Table for all event types except a command to cancel an existing fire command, which results in an entry being removed from the table. The table is kept sorted, first on fire unit number, second on weapon number, to facilitate searching by the Ground Fire Module, which implements the commands. Fire commands scheduled

for a given minute are stored in the Fire Control Table at the beginning of that minute's activities, before any Ground Fire processing has been initiated. When the Ground Fire Module is called, all entries in the Fire Control Table are processed before automatic fire allocation processing for the respective weapon types is performed to give the fire commands priority. Weapons not under command and control are processed according to the automatic allocation logic. When all weapon processing has been completed, the table is updated by reducing the duration one minute, removing expired entries, and re-sorting, if applicable.

Support fire commands directed against area targets require an additional piece of information saved in the Fire Control Table. The location of the area target is stored in the Impacting Fires Table. The position (or pointer) within this table is stored in the Fire Control Table, and is associated with the target.

4.7.2 Programmatical Description

The Fire Control Table, as noted above, is comprised of two arrays. These arrays are depicted as follows:

	IFIROVRD(I)	ITGTLST(I,1)	ITGTLST(I,2)		ITGTLST(I,8)	ITGTLST(I,9)	ITGTLST(I,10)
1							
2							
3							
.							
.							
.							
100							

An entry in the table consists of a single entry in the IFIROVRD array, say IFIROVRD(I), and up to eight entries in the array ITGTLST, say ITGTLST(I,1),...., ITGTLST(I,8), where index I is identical in both arrays. ITGTLST(I,9) and ITGTLST(I,10) represent 8 byte-size pointers into the temporary impacting fires array (IMPTEMP(100)) for the eight targets, respectively. Non-area targets do not use the pointer. The pointer ranges in value from 1 to 100. The format of the arrays is depicted in Table 4-3 .

When a fire control command is issued via the Fire Control Menu, a 64-word event notice is stored on the background events file along with its scheduled time of occurrence. When the simulated time occurs, the Events Module, in particular subroutine PPEVENT, calls subroutine FIREVNT for each fire control event to be processed. The event notice is depicted in Table 4-4 .

Subroutine FIREVNT merely determines whether to call FIRSORT once (if the entered command was for a single, specified weapon), or repeatedly (once for each weapon in the specified unit), if "all weapons" was specified. Individual

targets for "rounds per minute" commands are also checked for duration, since not all target durations are necessarily identical for this type of fire command. If a "cancel fire mission" is indicated by the subevent type (=8) of the event notice, the designated command is removed from the fire control table and the table is squeezed. Next, the table is checked for availability, and error-checking is done on the validity of the unit and weapon numbers specified in the event notice. The duration of the command is forced to be within 1 and 255 minutes (a duration of zero is set to one; a duration of greater than 255 is set to 255). If a "rounds per minute" command has been specified, the total number of rounds allocated is checked against the weapon's max firing rate, and a rejection alert is issued if that rate is exceeded. The table is searched for a match on unit/weapon. If no entry exists for the specified unit/weapon, the command is inserted in the table, sorted primarily by unit, secondarily by weapon. If a command already exists for the unit/weapon , it is overridden unless the new command is a "rounds per minute" type (subevent type 1 or 3). In this case, if the entry already in the table is for "percent of fire" (subevent type 2 or 4), the new command is ignored; if the entry already in the table is for "rounds per minute," the new targets are added to the targets already being fired at, unless the max firing rate is exceeded, in which case the new command is rejected. If the new command is "cease fire" (subevent type 1), the table entry is altered and processing of the command is completed. Otherwise, the targets of the new command are checked one at a time to determine if they are within min and max range constraints of the weapon. If a target does not meet the constraints, it is rejected, an alert is issued and processing continues.

If fire is directed from a support fire weapon against an area target (target number 201 as indicated by the event notice), subroutine FIRSORT searches the IMPTEMP array for an available slot in which to store the X,Y position of the target point. The index of this slot in the IMPTEMP array is stored in the appropriate byte of words 9 and 10 of the Ith row in array ITGTLST. Eight bytes exist in words 9 and 10, corresponding to the eight possible targets associated with fire command I.

An entry formatted according to Table 4-3 is the result of processing a fire control event.

When the Ground Fire Module is called later in the time step, the Fire Control Table contains all those commands which are to be executed this time step. These commands were entered into the table either at the beginning of the current time step, or had been entered previously and have not yet expired. The Ground Fire Module, for a given weapon type, processes all table entries first by searching

for a match on unit/weapon in the first half of the IFIROVRD entries. If less than 100% of a weapon's firepower in a unit is specified by a command, the remainder is allocated according to the automatic allocation logic. "Cease fire" commands, of course, prohibit firing of a weapon.

After both direct (and indirect) and support fire weapon processing has been completed, the Fire Control Table is aged by decrementing the duration in array IFIROVRD and, if a "rounds per minute command" is being aged, the individual durations associated with each target (since they are not necessarily identical). If a duration in IFIROVRD reaches zero, the entire entry is removed from the table, including the data in array ITGTLST. The table is squeezed following the expiration of a command to keep it compact. If an individual target duration expires, but the command still is in effect against other targets, the expired data from ITGTLST is removed and that row of the array is squeezed laterally.

Table 4-3 . Fire Control Arrays

IFIROVRD (I), (I=1,100)

Bits	0-------7	8-------15	16-------23	24-----27	28---31
	A	B	C	D	E

A - Unit Number (1-100)
B - Weapon Number (1-80)
C - Duration (1-255)
D - Subevent Type (0-7)
  0 = percent of fire, suppression
  1 = rounds/minute, suppression
  2 = percent of fire, no suppression
  3 = rounds/minute, no suppression
  4 = not used
  5 = not used
  6 = not used
  7 = cease fire
  8 = cancel fire mission
E - number of targets

ITGTLST (I,J), ((I=1,100), J = 1,8)

Bits	0-------9	10-------17	18-------31
	A	B	C

A - Target Number
  0 XY Point
  1-100 Unit Number
  101-200 Bridge Number + 100
  201-700 Road Segment Number + 200
B - Duration in minutes
  (used by RPM only)
C - Number of rounds to fire, or
  percent of fire * 100

ITGTLST (I,J), ((I=1,100), J=9,10)

Bits	0-------7	8-------15	16-------23	24-------31
	A	B	C	D

A,B,C,D - Pointers to the impacting fires array (IXYIM), one for each of the 8 targets. Word 9, Byte A, corresponds to target 1, ITGTLST (I,1), etc.

Table 4-4 . Fire Event Notice Format

WORD	
1	=10 (fire event type)
2	Subevent type
3	Fire unit #
4	Weapon #
5	Duration (minutes)
6	Number of targets
7	Target #, 1st target
8	%of fire (x100) or rounds allocated against 1st target
9	X location, 1st target
10	Y location, 1st target
11	Target #, 2nd target
12	% of fire (x100) or rounds allocated against 2nd target
13	X coordinate, 2nd target
14	Y coordinate, 2nd target
.	.
.	.
.	.
35	Target #, 8th target
36	% of fire (x100) or rounds allocated against 8th target
37	X coordinate, 8th target
38	Y coordinate, 8th target
39-63	NOT USED
64	Instructor #

4.8 WEAPONS EFFECTS LOOK-UP TABLE

The Weapons Effects Look-up Table consists of the following array:

IFNTB(1000,2)

4.8.1 English-language Description

The Weapons Effects Look-up Table is used in conjunction with the Weapons Effects Coefficient Set Table to perform casualty assessment for a specified number of rounds of a single weapon type in one firing unit against one opposing target unit. All ground weapons, whether allocated automatically or via command/control, compute weapons effects this way. The submodule logic examines each of the four personnel vulnerability classes and all equipment types existing in the target unit and assesses casualties separately on each. An entry must exist in the weapons effects table in order for casualties to be assessed on any personnel class or equipment as a result of a particular weapon firing. The entry consists of a code word and a data word.

The code word contains the following information:

- operational state
- weapon number
- mode of fire
- either target personnel vulnerability class or equipment number.

The process of computing a weapon's effects against each individual target element in a target unit is begun by finding a matching code word in the Weapons Effects Look-up Table. If none is found, the weapon has no effect on the target element. If a match is found, the corresponding data word is extracted from the table, containing:

- the weapon effects function to be used
- the allocation criterion to be used, if function 2 is the selected function, for aimed fire against equipment
- the coefficient set to be used with the function from the Weapons Effects Coefficient Sets Table.

The weapons effects function is one of the following:

Function 1: $E_k = V * P_k \; [1 - \exp(-k_4/D^2)]$

Function 2: $E_k = V * P_k$

Function 3: $E_k = V * N * L/A$

Function 4: $$E_k = \frac{k_1 \, N \begin{bmatrix}\text{Unit personnel losses during time step due to effects} \\ \text{of given weapon type against vulnerability class } k_2\end{bmatrix}}{[\text{No. of target unit personnel in vulnerability class } k_2]}$$

$$k_1 \, N \left[\frac{[\text{No. of target weapon crews lost}]}{[\text{No. of target weapon crews in target unit}]}\right]$$

Function 5: $E_k = N * [1 - \exp(-V * L/A)]$

Function 6: $E_k = N * [1 - \exp(-V * L * F_\alpha/A)]$

Function 7: (air defense) $E_k = 1. - (1. - P_k * P_{k/H}/NWPNS) * * V$

where:

k_1 = Expected number of target weapons lost to fire for each weapon crew lost. This kind of factor can also be used with nonweapons that are target elements.

k_2 = Personnel vulnerability class of crews operating the target equipment. $k_2 = 0$ implies that more than one personnel vulnerability class is involved, and consequently, E_k is computed, using the total numbers of personnel and personnel losses in the target unit.

k_3 = The minimum range to be used in function 1. This function is commonly used to estimate the effect of aimed fire subject to normally distributed aiming and dispersion errors.

$$k_4 = \frac{(\text{presented area of target element in sq meters} \; . \; 10^6}{(\text{variance of aiming and dispersion error in mils})^2 \; . \; 2\pi}$$

D = Sup $\{k_3, R\}$

P_H = Probability of a hit with a single round

$P_{k/H}$ = Probability of a kill given a hit for a single round

V = Volume of fire in rounds

N = Number of target elements

L = Lethal area = $a + bR + cR^2$

A = Target unit area fired into = length * width * fraction fired into

R = Range between fire and target units

F_α = Fraction of rounds falling inside area fired at

E_k = Expected casualties

P_k = Probability of kill for a single round.

If function 2 is used to assess casualties, the allocation criterion indicates which of four allocation methods should be used in directing the fire against the different types of equipment in the target unit that are eligible for the fire. The four allocation methods are:

Criterion No.	Allocation Method
1	Allocation of fire against an equipment type is proportional both to the weighting factor indicating its importance as a target, and to the expected number of equipment casualties that would be resulted if all available fire were used against the equipment type.
2	All available fire is allocated to that equipment type having the greatest value of the product: (weighting factor indicating importance of equipment as a target) x (expected equipment casualties if all available were used against the equipment type).
3	Allocation of fire against an equipment type is proportional both to the weighting factor indicating its importance as a target, and to the number of items of this equipment present in target unit J.
4	All available fire is allocated to that equipment type having the greatest weighting factor indicating its importance as a target.

The current scenarios utilize just two of the four variables of the code word:

- weapon number
- either target personnel vulnerability class or equipment number.

Although a total of seven weapons effects functions are defined in the code, six for ground fire and one for air defense, just five of them are utilized by the current scenarios, four ground fire functions and the one for air defense. The four ground fire functions used are numbers 2, 3, 5, and 6 listed above.

4.8.2 Programmatical Description

The array IFNTB(I,J) is dimensioned 1000x2, allowing for 1000 2-word entries, the first word of which is a code word, the second a data word. The table is loaded at model initialization by subroutine WEFINP. Although the array can hold a capacity of 1000 entries, the current scenarios supply 539, 76 of which are against personnel classes, 463 against equipments. After initial loading, the table is never changed.

Each time step, for every weapon in a firing unit allocating rounds of fire against an enemy target unit, either automatically or interactively, a weapons effects calculation is necessary. The calculation begins with a call to subroutine WPNFIR from either of the following routines:

ORGFIR	(direct and indirect fire weapons)
SPTALO, GENFIR	(support fire weapons against unit targets)
SETIMP1	(support fire weapons against points)
SETIMP2	(support fire weapons against bridges and roads).

WPNFIR strips the code word into its four variable components. The code word is formatted as SSXXYZZ, where

SS = operational state
XX = weapon type
Y = mode of fire
ZZ = target personnel vulnerability class or equipment class.

The values for SS and Y are set to 0 for all entries to eliminate the need to match operational state and mode of fire. Therefore, a match on weapon type and either target personnel vulnerability class or equipment class suffices to determine the corresponding data word. The entries in the table are ordered. All entries for which ZZ is a personnel vulnerability class are first; those for ZZ as an equipment number follow. Subroutine WPNFIR first loops over all personnel vulnerability classes in the target unit, searching the first part of the table only. When a match is found for the particular class and firing weapon, the variables are extracted from the corresponding data word. The data word is formatted as FADDD, where:

F = weapon effects function to be used

A = allocation criterion (for function 2 only)

DDD = effects data coefficient number.

Subroutine EFFNS contains the algorithms for each of the seven weapons effects functions. The coefficients for the designated function are extracted from array EFFDAT (Weapons Effects Coefficient Sets Table) using DDD as an index. Casualties are assessed for the particular personnel vulnerability class and stored. The next class is assessed until all personnel vulnerability classes have been exhausted in the unit. Then WPNFIR loops over all equipment types in the target unit one by one as they appear on the target unit's equipment list. The second half of the table containing entries against equipment is searched for a matching code word. As before, if none is found, the firing weapon has no effect against the equipment being examined. If a match is found, the variables are extracted from the corresponding data word and casualties are assessed against the equipment type. The next equipment is assessed similarly until all equipment types in the unit have been exhausted.

An example of the use of the Weapons Effects Look-up Table follows. The data used is from the FEBA Gold scenario.

Example

Suppose a Blue 155 MM howitzer (weapon number 35) is firing at Red unit C/TK/107 (unit number 16) in the first minute of a FEBA Gold exercise. The Red unit, C/TK/107, has the following personnel distribution and equipment list:

Personnel Vulnerability Class	Amount	Equipment List	Amount
1	1	10 (tanks)	10
2	4	21 (trucks)	1
3	4	1 (rifles)	46
4	37		

The following excerpts include all the entries in the Weapons Effects Look-up Table for weapon number 35 as a firing weapon (entries 1 thru 76 are against personnel as targets; entries 77 thru 539 are against equipment types as targets):

		SSXXYZZ	FADDD
(Personnel Section)	Entry #51	0035002	60037
	Entry #52	0035003	60039
	Entry #53	0035004	60041
(Equipment Section)	Entry #437	0035001	62157
	Entry #438	0035002	62157
	Entry #439	0035003	62157
	Entry #440	0035004	62157
	Entry #441	0035005	62158
	Entry #442	0035006	62004
	Entry #443	0035007	62158
	Entry #444	0035008	62159
	Entry #445	0035009	62161
	Entry #446	00350010	62161
	Entry #447	00350011	62103
	Entry #448	00350012	62162
	Entry #449	00350013	62163
	Entry #450	00350014	62113
	Entry #451	00350015	62165
	Entry #452	00350016	62113
	Entry #453	00350017	62113
	Entry #454	00350018	62108
	Entry #455	00350019	62108
	Entry #456	00350020	62108
	Entry #457	00350021	62164
	Entry #458	00350044	62004
	Entry #459	00350045	62164
	Entry #460	00350046	62164
	Entry #461	00350091	60183
	Entry #462	00350092	60167
	Entry #463	00350095	60166
	Entry #464	00350096	60175
	Entry #465	00350097	60171

Subroutine WPNFIR would loop through personnel vulnerability classes 1, 2, 3, and 4. Since there is no entry against personnel vulnerability class 1 (ZZ = 01), the effect is zero. For personnel vulnerability classes 2, 3, and 4 (ZZ = 02, 03, 04), function 6 is used (F = 6) with coefficient sets 37, 39, and 41 (DDD = 037, 039, 041), respectively. Similarly, entries 446, 457, and 437 are used for equipments 10, 21, and 1, respectively. The equipments are processed in the same order as they appear in the equipment list. Again function 6 is used, with coefficient sets 161, 164, and 157, respectively.

4.9 UNIT TYPE MAX RANGE TABLE

The Unit Type Max Range Table consists of the following array:

IFRNG(20, 2, 2)

4.9.1 English-language Description

The Unit Type Max Range Table is used primarily to reduce the computer time required to search for potential target units of direct and indirect fire weapons being automatically allocated. For each Red unit type a maximum direct fire and a maximum indirect fire range is specified. For direct fire weapons each range is determined by taking the maximum range of all direct fire weapons used by any unit of that type. When a direct fire weapon is processed, target units outside this range will be immediately eliminated from consideration before any detailed target unit processing is conducted. The indirect fire max range is treated similarly. Blue direct and indirect fire weapons are treated similarly.

Weapons having a fire command being executed this time step do not use this table.

4.9.2 Programmatical Description

Subroutine FIRELG determines which opposing target units are eligible for detailed fire allocation processing for direct and indirect fire weapons. The first test for a target unit is whether it is within the maximum firing range for the firing unit type.

The distance between the two units is computed via a call to PPDIST. This distance is compared to the value IFRNG(IUT,KFICTR,ICOL), where

IUT = unit type of the firing unit
(there can be up to 20 unit types)

KFICTR = 1 direct fire weapon
2 indirect fire weapon

ICOL = 1 red
2 blue.

If the test is passed, the target unit is processed further; otherwise, it is eliminated.

The IFRNG data is read into the system along with unit type inputs. An example of the IFRNG data input taken from the FEBA Gold scenario follows.

Example

Unit type 2, "ARMOR", has the following IFRNG inputs:

Red, direct fire: IFRNG (2, 1, 1) = 2000

Red, indirect fire: IFRNG (2, 2, 1) = 0

Blue, direct fire: IFRNG (2, 1, 2) = 2000

Blue, indirect fire: IFRNG (2, 2, 2) = 0

The following red and blue units are designated unit type 2:

Unit	Firing Weapons
C/TK/107	T-62/115 (tank), AK-47 (rifles)
1, 3/A/2-4 1, 2, 3/B/2-4 1, 2, 3/C/2-4	M-60 105 MM (tank)
2/A/2-4	M-16 5.56 MM (rifles) M-60 105 MM (tank)

The list of weapons contains all direct fire weapons with the following maximum ranges:

Weapon	Range (meters)
T-62/115 (red)	2000
AK-47 (blue)	400
M-60/115 (blue)	2000
M-16 5.56 MM (Blue)	460

Since 2000 is the maximum direct fire range for both red and blue, the corresponding elements of array IFRNG are set to 2000. Since no indirect fire weapons are assigned to unit type 2, the corresponding elements of IFRNG are set to 0.

4.10 FIRE SUPPORT TABLE

The Fire Support Table consists of the following array:

IFSTAB(80,2)

4.10.1 English-language Description

The Fire Support Table is used to indicate how support fire weapons are to be automatically allocated against opposing target units. Each entry in the table consists of one code word and one data word. The code word consists of the following:

- Weapon (equipment) number
- Operational state of the firing unit
- Support fire unit number.

The data word consists of the following:

- Indicator to interpret the mode distribution vector either as 1) fraction of total pieces of equipment to be assigned to each mode or 2) weighting factor to be applied to target weights found in each mode.
- Flag indicating whether engagement regions should be defined relative to friendly or enemy FEBA's.
- Flag indicating whether close-support fire should be directed against individual target units or uniformly over the whole close support region.
- Index number of the fire support target band to be used.
- Flag indicating whether the target band allocation factor should be used as 1) a fractional allocation of pieces to the band or 2) a weighting factor to be used for targets in the band.
- Flag indicating if general support fire is either 1) restricted to targets within the Y-bounds associated with the Fire Support Table entry or 2) directed at all target units within range.
- Mode distribution vector number to be used for the equipment being processed.

Two additional data words defining sector bounds are associated with these table entries, as alluded to in the sixth item under data word above. The first word specifies the lower Y-bound of the sector, the second the upper Y-bound. Support fire weapons for which a current command exists do not use this table.

Every support fire weapon not under command/control is processed in a time step by initially searching the code word portion of the Fire Support Table until a match is found. Having found a match, the seven data items listed above and the associated sector bounds are extracted and stored for use.

Assume support fire weapon type J in unit I is being processed by the Support Fire Submodule of the Ground Fire Module. Before processing any individual weapon, all opposing units are geographically categorized with respect to existing engagements. This categorization determines which mode of support fire each opposing unit is eligible for. The mode distribution vector number is extracted from the Fire Support Table to determine how many weapons are to fire in each mode. There are two ways to interpret this vector, depending on item 1 of the Fire Support Table data word, as follows:

Fire-Support Mode i	Interpretation of Element i of Mode Distribution Vector	
	T = 1	T = 2
1,2 (non-firing)	Fraction of all available weapons (type J in unit I) allocated to the mode. (Interpretation is identical to that for other equipment categories, such as direct fire weapons.)	Same as for T = 1.
3 (close support, normal)	Fraction of all available weapons (type J in unit I) allocated to the mode. (Interpretation is identical to that for other equipment categories.)	Special mode weighting factor (to be combined with other target-unit-weighting factors) for targets associated with modes 3 and 4.)
4 (close support, final protective fires)	Fraction of all available weapons (type J in unit I) allocated to the mode. (Interpretation is identical to that for other equipment categories.)	Fraction of all weapons allocated to modes 3 and 4 combined that are to operate in mode 4.
5,6,7,8 (other firing modes)	Fraction of all available weapons (type J in unit I) allocated to the mode. (Interpretation is identical to that for other equipment categories.)	Special mode weighting factor for targets associated with the mode.

If the mode distribution vector expresses weighting factors (i.e., T = 2), then the weight specified for a given mode of fire is combined with all other weights assigned a target in that mode (e.g., target value, special threat, target acquisition) to give the target a total fire-support allocation weight. The available fire-support weapons (type J in unit I) are then allocated against all eligible targets strictly on the basis of this total allocation weight.

Two of the eight modes represent fire at enemy units that pose a direct threat to friendly forces in an engagement. The areas into which close-support fire is directed are defined by rectangular regions in established engagements. Each region includes the enemy FEBA and a number of enemy units, the number being dependent on the depth of fire desired. Fire can be directed at targets within each region or distributed uniformly over the entire region.

Item 3 of the data word determines which method is to be used. Such a region is defined relative to either the Friendly or Enemy FEBA, whichever is indicated by item 2 of the data word.

Modes 6 and 8 are general fire support modes, mode 6 reserved for units in operational groupings and mode 8 reserved for all other units. Sets of fire-support target bands used with modes 6 and 8 are defined by input. The appropriate set of bands to use with weapon type I in unit I depends on the unit's code word match and is specified by data word item 4 of the selected entry in the Fire Support Table. Associated with each set of bands is a set of band priorities and band allocation factors. See a complete discussion in section 3.0 under the term "bands".

Item 5 of the data word is best described by an example. Suppose now that n weapons have been allocated to mode 6 and that mode 6 targets have been partitioned into fire-support target bands. If item 5 is 1, then these n weapons will be allocated to the various bands in the order of the given band priorities, and the fraction allocated will be the value of BANDAL or the fraction of the n weapons remaining to be allocated, whichever is less. The sum of the BANDAL values for a set of bands should ordinarily total more than 1.0. If item 5 is 2, band priorities are not used, and the values of BANDAL become a set of band weighting factors. These factors are combined

with all other weights assigned the mode 6 targets, and fire from the n available weapons is then allocated to these targets strictly on the basis of target weight. Mode 8 targets are treated in a similar way.

A fire-support unit is often required to restrict its fire to the direct support of a particular combat force or to targets located within a particular sector of the battlefield. In the automatic support fire allocation, such a restriction is imposed by specifying the boundaries of a sector and requiring unit I to limit its fire of weapon J to targets lying within this sector. Sector boundaries are given as two lines parallel to the x-axis. Options are available (1) to fire only at targets located within the given sector or (2) to fire at close-support and interdictory-fire targets within the sector, but fire at other classes of targets wherever they may be found. The desired option is indicated by item 6 of the data word. The sector into which unit I is allowed to fire may be changed merely by changing the operational state of unit I, causing a match on a different code word of the Fire Support Table.

4.10.2 Programmatical Description

The IFSTAB array is dimensioned 80x2 to accommodate eighty two-word entries, the first word being a code word and the second being a data word. Data is read into the array at system initialization and not altered during the exercise of the model. The format of the two words is defined as follows:

IFSTAB(I,1) = AABBCC,

where:

AA = weapon type (00 = > any type).

BB = operational state of support-fire unit (00 = > any state).

CC = unit number (00 = > any number).

IFSTAB(I,2) = TUVWXYZZ,

where:

T = interpretation of mode distribution vector to use.

1 = > interpret the vector in exactly the same way as with other equipment categories (i.e., fraction of fire in each mode).

2 = > interpret the entries in the vector as additional weighting factors applied, for purposes of fire allocation, to targets fired at under the various fire-support modes. See English-language discussion.

U = 1, if engagement regions (containing close-support and interdictory-fire targets) are defined relative to friendly FEBA's.

= 2, if the regions are defined relative to enemy FEBA's.

V = 1, if fire in close-support regions is directed at individual target units.

= 2, if fire is directed uniformly over the whole close-support region.

W = Index number of the fire-support target band set to use (1-9).

X = 1, if the target band allocation factor (BANDAL), for weapons firing in mode 6 or 8, represents the fraction of these weapons to be used against the given band.

= 2, if the band allocation factor is an additional weighting factor for targets in the band (used for allocation among the bands).

Y = 1, if general-support fire (modes 6-8) is restricted to targets in the assigned fire-support sector (defined by IYBDS).

= 2, if general-support fire may be directed at all target units within range.

ZZ = number of the mode distribution vector to use.

The format of the code word places a two-digit constraint on the three variables: weapon number; unit operational state; unit number. The

dimension of the arrays corresponding to these variables is 80, 100*, and 100*, respectively. Obviously, operational state 100 and unit number 100 cannot be used with this format.

A related array, IYBDS(80,2), which defines lower and upper sector boundaries, has its primary index derived for it by the IFSTAB code word match. The index, I, for which the match occurs is used for both the IFSTAB data word and the corresponding sector bounds, IYBDS(I, 1 and 2).

Data element W of the IFSTAB data word is the first index into the array BANDAL(9,4), where W is the number of the band set being used. Since W is a single digit code, the code word format limits the number of different band sets that can be used in the model to 9, which is used currently.

The number of mode distribution vectors is limited by the data word element ZZ to two digits. The mode arrays are presently dimensioned 80, although just 43 are presently defined.

Subroutine SPTALO is the sole user of the IFSTAB array. Information is stripped from the code and data words by repeated divisions of powers of 10, identical to the procedure described in the Mode Selection Code Table. The seven data items are then used as either flags or indexes into other arrays, as described individually above. Only support fire weapons allocating fire automatically, as opposed to being under a fire command, use the data in IFSTAB.

---

*Several arrays relating to these concepts are dimensioned greater than 150, where those array elements having an index greater than 100 are for either operational groupings or adjacent units, both of which are treated differently than units.

4.11 IMPACTING FIRES TABLE

The Impacting Fires Table consists of the following arrays:

IXYIM (100)
IXYIMPTR (10)

4.11.1 English-language Description

The Impacting Fires Table is used to store X,Y locations for which an impacting fires symbol is to be displayed the next time step. These X,Y locations qualify for such a symbol when either one of two events occurs:

1) a support fire weapon fires at a designated point target as a result of a fire control command having been implemented;

2) an air unit delivers ordnance as a result of an air strike mission.

The impact point of the ordnance from either artillery or air is stored in the main array, IXYIM, in the first available slot. An additional impacting fires indicator is displayed for air-delivered ordnance. A supplementary array of pointers, IXYIMPTR, is used to indicate which of the locations stored in the main array are delivered from the air.

4.11.2 Programmatical Description

When a fire command is generated by a controller by means of the fire control menu, it is stored on the background events file until its scheduled time of occurrence. When that time arrives, subroutine FIREVNT is called to process the event, and FIREVNT calls subroutine FIRSORT to perform most of the processing. One of FIRSORT's functions is to check if the command involves a support fire weapon firing at a point target. If so, the X,Y location must be saved for the Ground Fire Module in array IMPTEMP (100). The first available slot in the IMPTEMP array, indicated by a zero, is used to store the point target location. The pointer into the array is stored in the appropriate byte of the eight bytes of ITGTLST (I,9) and ITGTLST (I,10) for fire command I (see discussion of Fire Control Table). Before storing the X and Y locations, the two LSB's are shifted off both values (equivalent to dividing by 4 and truncating).

The resultant X-value is stored in the left half of the available IMPTEMP word, and the resultant Y-value is stored in the right half. If the firing unit is Red, the entire word is set to negative to indicate that a Red symbol should be drawn; if Blue, the word remains a positive value.

When the Ground Fire Module processes support fire weapons primarily in subroutine SPTALO, command and control events are processed first. A weapon under command and control is processed one target at a time. Each target is checked to determine if it is a point target, denoted by a zero in the appropriate ITGTLST target number entry. When a point target is found, subroutine SETIMP1 is called to determine if the point falls within any unit's area. If so, casualty assessment is performed for the unit. The IMPTEMP entry is then copied into the first available (=0) slot in array IXYIM, which gets accessed by the foreground programs for display the next time step.

Similarly, support fire weapons firing at a bridge or road segment call SETIMP2 and store the X,Y location of the designated bridge or road segment into the first available entry in IXYIM in the same format as point targets. A bridge target is denoted by a number from 101 to 116, where 101 represents bridge #1, ···, 116 represents bridge #116. A road segment target is denoted by a number from 201 to 700, where 201 represents road segment #1, ···, 700 represents road segment #500.

Air unit interaction with ground units is performed prior to processing of the Ground Fire Module. An air unit delivering ordnance against a target computes the point of impact of the ordnance regardless of the type of target type. The impact point is then stored in the first available slot in array IXYIM in the packed format. To distinguish air-delivered impacts from ground-delivered, a supplementary array, IXYIMPTR (10), one entry for each possible air unit up to the maximum of 10 allowed in the model, contains a pointer to the location of the X,Y impact position in the IXYIM array. The ten pointers in array IXYIMPTR correspond to the 10 possible air units. These pointers are also accessed by the foreground programs in order to display air unit symbols along with the impacting fires symbols, distinguishing them from support fire impacts.

An example of the packing scheme follows. If a Red air ordnance delivery occurs at X = 49159, Y = 16385 by the first air unit, this point is packed into the IXYIM array. Assume IXYIM and IXYIMPTR exist as depicted below prior to storing the above information.

IXYIM (binary representation)

1)	0 0 1 0 0 1 0 1 1 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 1 1 1
2)	0 1 0 1 1 1 0 1 0 0 0 1 1 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 0 1
3)	0 ——————————————————————— 0
.	
.	
.	
100)	0 ——————————————————————— 0

IXYIMPTR (decimal representation)

1)	0
2)	0
.	.
.	.
.	.
10)	0

The binary representation for X = 65543 is

00000000000000010000000000000111

and for Y = 49153 is

00000000000000001100000000000001

Both binary representations are shifted right two bits to produce, respectively,

0 ———————— 0010000000000001

0 ———————— 0001100000000000

Then, X is stored in the left half, Y in the right half of the first available slot in IXYIM, which is IXYIM (3), to yield

01000000000000010011000000000000

Since the air unit is Red, the number is negated (two's-complemented), represented by

10111111111111101101000000000000

A 3 is placed in IXYIMPTR (1) to indicate that the third impacting fires entry (IXYIM(3)) is for an air delivery. Arrays IXYIM and IXYIMPTR appear as follows after the above information is entered:

IXYIM (binary)

1)	0 0 1 0 0 1 0 1 1 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 1 1 1
2)	0 1 0 1 1 1 0 1 0 0 0 1 1 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 0 1
3)	1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
4)	0 —— 0
.	
.	
.	
100)	0 —— 0

IXYIMPTR (decimal)

1)	3
2)	0
.	.
.	.
.	.
10)	0

4.12 CHANGE OF STATE TABLE

The Change of State Table consists of the following arrays:

IOPTB1 (220,2)
IOPTB2 (220)
DECVAL (150)

4.12.1 English-language Description

The capability is built into the model to enable units to automatically change their mode of operation (operational state) if certain criteria are met. These criteria involve simulated time from the start of the exercise, unit location (both absolute and relative to friendly and enemy forces), actions of other units, casualties sustained, force ratios, unit suppression, levels of ammunition and equipment, and other factors.

The Change of State Table contains code words and data words which, together with program logic evaluating the criteria, implement this capability. A typical entry in the table consists of two packed code words and a packed data word. Every time step, the table is searched for every active (non-defunct) unit not under maneuver control and not in a delayed state as a result of an obstacle encounter. The unit attempts a match on the following unit attributes: unit type; operational group number; Red or Blue; movement code; current operational state; unit number. If a match is found, the packed data word is broken into its several components:

1) change of state criterion number;
2) line number of the first associated data value (in array DECVAL);
3) new operational state to which the unit is to change;
4) code indicating whether or not a unit associated with an operational grouping (OG) should have the OG.

The change of state criterion number indicates which of the 85 criteria actually coded in the model is to be evaluated. The criteria definitions are listed in Programmatical Description. One or more data values are required to evaluate most of the criteria. The first data value required for a criterion evaluation is specified by item 2 of the data word. The new operational state to which the unit is to move and the OG indicator are both implemented if the specified criterion is satisfied. Any unit for which no match exists in the code portion of the table, or for which a matching code is found but the indicated criterion is not satisfied, is not affected by the above procedure.

For a unit having a matching code word for which an operational state change was performed, a second search of the table may be done using the new operational

state of the unit to attempt a code word match. If the second search is successful, the unit may change operational state a second time, but no more. Any unit undergoing a change in operational state subsequently has its movement code checked to see if a corresponding change is required.

4.12.2 Programmatical Description

The Change of State Table is initialized along with the rest of the data base at system start-up. Subroutine COSINP reads all of the desired data into the Change of State Table arrays: IOPTB1, IOPTB2, DECVAL. These values are not altered throughout an exercise, with the exception that an entry may be deleted. Every time step, subroutine OPPLAN is called by the main math model driver program. FORMAIN, to inspect all active units (ISTATU (IU) ≠ -1) not under maneuver control (MOVECC (IU) = 0) and not delayed as a result of an obstacle encounter (IOBSTATU (IU) ≠ 2). Units passing through the above screening are sent to subroutine CHGOPN, where a search is performed on arrays IOPTB1 and IOPTB2. Array IOPTB1 contains up to 220 two-word entries, entry I consisting of IOPTB1 (I, 1) and IOPTB1 (I, 2). An entry consists of a 11-digit number of the form UUVVWXXYYZZ,

where

UU	=	unit type (00 => any type)
VV	=	operational grouping number (00 => disregard op group factor, 99 => any unit not in an op group)
W	=	1 or 2 (red or blue) (0 => either)
XX	=	movement code (00 => any code, 88 => any engaged code (1-4), 99 = any non-engaged code (15-16))
YY	=	current operational state (00 => any op state)
ZZ	=	unit number (00 => any unit number)

If a match is found on the 10-digit code word, the corresponding entry in array IOPTB2, represented by IOPTB2(I), is used to determine if a change in operational state should be implemented for the unit. IOPTB2(I) is a signed 8-digit number of the form ± AABBBCCD,

where

AA	=	change-of-state criterion number (see next page)
BBB	=	line number of the <u>first</u> associated data value
CC	=	new operational state
D	=	code to tell whether unit should leave its operational grouping (0 => no, > 0 => yes).

Subroutine CHGCRT is called to evaluate the criterion number specified in the AA portion of the word. Typically the result is compared to a threshold value(s) specified in DECVAL (BBB), where BBB is a 3-digit pointer into the DECVAL array

(if more than one value is used for the threshold, BBB points to the first and if a second value is required, it is put in the table as the next consecutive entry). If the test is passed, the operational state of the unit becomes that value specified in item CC of the IOPTB2 word. Finally, code D indicates whether or not the unit should leave its operational grouping.

There are 85 different criteria evaluated in subroutine CHGCRT. These criteria are hard-coded in the subroutine. A list of the 85 criteria follows, where the unit being processed is referred to as "unit I," and the threshold value is "IDATA" (second threshold value, if applicable, is "IDATA1"):

(1) time ≥ IDATA
(2) time = IDATA
(3) x coordinate of unit I ≥ IDATA
(4) x coordinate of unit I < IDATA
(5) x coordinate of unit I (rounded to nearest 100) = IDATA
(6)* Distance from unit I to nearest enemy unit <u>in same engagement</u> ≤ IDATA
(7)* Distance from unit I to nearest enemy unit <u>in same engagement</u> > IDATA
(8) Distance from unit I to nearest enemy unit ≤ IDATA
(9) Distance from unit I to nearest enemy unit > IDATA
(10)* Distance between unit I and enemy FEBA ≤ IDATA
(11)* Distance between unit I and enemy FEBA > IDATA
(12)* Distance between unit I and friendly FEBA ≤ IDATA
(13)* Distance between unit I and friendly FEBA > IDATA
(14)* x coordinate of the location of the enemy FEBA ≤ IDATA
(15)* x coordinate of the location of the enemy FEBA > IDATA
(16)* x coordinate of the location of the friendly FEBA ≤ IDATA
(17)* x coordinate of the location of the friendly FEBA > IDATA
(18)* Distance between the opposing FEBAs ≤ IDATA
(19)* Distance between the opposing FEBAs > IDATA
(20) Degree of suppression of unit I ≥ IDATA
(21) Degree of suppression of unit I < IDATA
(22) Not used
(23) Unit I is below critical minimum level for at least one principal ammunition type
(24) Unit I is above critical minimum level for at least one principal ammunition type
(25) Unit I is below critical minimum level for all principal ammunition types

(26) Unit I is above critical minimum level for all principal ammunition types

(27) $$1 - \frac{m_I(t)}{m_I(0)} \geq \text{IDATA (fraction of men lost in unit I)***}$$

(28) $$1 - \frac{n_{I^1}(t)}{n_{I^1}(0)} \geq \text{IDATA (fraction of first equipment type in unit I that is lost)***}$$

(29)* $$1 - \frac{\Sigma m_J(t)}{\Sigma m_J(0)} \geq \text{IDATA}$$ (for friendly units J in the same engagement as unit I with a status code equal to 1) ***

(30) $$\frac{1}{m_I(0)} \quad \frac{\delta m_I}{\delta t} \geq \text{IDATA ***}$$

(31) $$\frac{1}{m_I(0)} \quad \frac{\delta m_I}{\delta t} < \text{IDATA ***}$$

(32) $$\frac{1}{n_{I_1}(0)} \quad \frac{\delta n_{I_1}}{\delta t} \geq \text{IDATA ***}$$

(33) $$\frac{1}{n_{I_1}(0)} \quad \frac{\delta n_{I_1}}{\delta t} < \text{IDATA ***}$$

(34)* $$\frac{1}{\Sigma m_J(0)} \quad \frac{\Sigma \delta m_J}{\delta t} \geq \text{IDATA}$$ (for friendly units J in the same engagement as unit I with a status code equal to 1) ***

(35)* $$\frac{m_I(0) - m_I(t)}{m_I(0)} - \frac{\text{distance to enemy FEBA}}{\text{unit I movement rate * } m_I(0)} \frac{\delta m_I}{\delta t} \geq \text{IDATA ***}$$

(36)* The Force Ratio** ≥ IDATA (for all friendly units J and enemy units K in the same engagement as unit I with a status code of 1)

(37)* The same as 36 except that all units J and K must be on the same side of the Engagement Axis as unit I

(38)* The Force Ratio** $<$ IDATA (for all friendly units J and enemy units K in the same engagement as unit I with a status code of 1)

(39)* The same as 38 except that all units J and K must be on the same side of the Engagement Axis as unit I

(40)* Position of unit IDATA $\leq$ IDATA1 from its enemy FEBA

(41)* Position of operational grouping IDATA $\leq$ IDATA1 from its enemy FEBA

(42)* Position of unit IDATA $>$ IDATA1 from its enemy FEBA

(43)* Position of operational grouping IDATA $>$ IDATA1 from its enemy FEBA

(44) Unit IDATA is in state IDATA1

(45) The forward-most unit of operational grouping IDATA is in state IDATA1

(46)* For the engagement of unit IDATA, the distance between the Red and Blue FEBAs $\leq$ IDATA1

(47)* For the engagement of unit IDATA, the distance between the Red and Blue FEBAs $>$ IDATA1

(48)* For the engagement of operational grouping IDATA, the distance between the Red and Blue FEBAs $\leq$ IDATA1

(49)* For the engagement of operational grouping IDATA, the distance between the Red and Blue FEBAs $>$ IDATA1

(50) Unit IDATA is engaged

(51) Operational grouping IDATA is engaged

(52) Unit IDATA is not engaged

(53) Operational grouping IDATA is not engaged

(54)* For the engagement of unit IDATA, the Force Ratio** in the Close Support Region $<$ IDATA1

(55)* For the engagement of unit IDATA, the Force Ratio** in the Close Support Region $\geq$ IDATA1

(56)* For the engagement of unit IDATA, the Force Ratio** in the Close Support and Interdiction Regions $<$ IDATA1

(57)* For the engagement of unit IDATA, the Force Ratio** in the Close Support and Interdiction Regions $\geq$ IDATA1

(58)* Same as 54 except use operational grouping IDATA

(59)* Same as 55 except use operational grouping IDATA

(60)* Same as 56 except use operational grouping IDATA

(61)* Same as 57 except use operational grouping IDATA

(62)* For the engagement of unit IDATA, the Force Ratio** in the left half-sector of the Close Support Region $<$ IDATA1

(63)* For the engagement of unit IDATA, the Force Ratio** in the left half-sector of the Close Support Region $\geq$ IDATA1

(64)* For the engagement of unit IDATA, the Force Ratio** in the right half-sector of the Close Support Region $<$ IDATA1

(65)* For the engagement of unit IDATA, the Force Ratio** in the right half-sector of the Close Support Region $\geq$ IDATA1

(66)* Same as 62 except operational grouping IDATA

(67)* Same as 63 except operational grouping IDATA

(68)* Same as 64 except operational grouping IDATA

(69)* Same as 65 except operational grouping IDATA

(70) Destination unit or operational grouping is defunct

(71) Destination engagement is defunct

(72) Rounds of support weapon fire/unit area falling on unit I $\geq$ IDATA

(73) Rounds of support weapon fire/unit area falling on unit I $<$ IDATA

(74) Status code of unit IDATA = 1

(75) Status code of unit IDATA $\neq$ 1

(76) x-coordinate of unit IDATA $\leq$ IDATA1

(77) x-coordinate of unit IDATA $>$ IDATA1

(78) x-coordinate of operational grouping IDATA $\leq$ IDATA1

(79) x-coordinate of operational grouping IDATA $>$ IDATA1

(80) This criterion sets the delay counter to IDATA unless the delay counter is not zero; normally, no change-of-state should be indicated

(81) This criterion decrements the delay counter of the unit by the magnitude of the time increment. A change-of-state will occur when the delay counter is less than or equal to the time increment

(82) Difference between x-coordinates of units I and IDATA $\leq$ IDATA1

(83) Difference between x-coordinates of units I and IDATA > IDATA1.

(84) (Distance between units I and IDATA) $\leq$ IDATA1.

(85) (Distance between units I and IDATA) > IDATA1.

*Unit I must be engaged or the test automatically fails.

**The Force Ratio is $\dfrac{\sum_J \sum_\beta u_\beta n_{J\beta} + \text{Measure of friendly support fire}}{\sum_K \sum_b u_b n_{Kb} + \text{Measure of enemy support fire}}$

where β ranges over all equipment types in the friendly unit J.

b ranges over all equipment types in the enemy unit K.

u is the equipment weight for equipment number n

***Variables used in defining criteria 27 through 35 are defined as follows:

$m_I(0)$ = number of personnel in unit I at time 0.

$m_I(t)$ = number of personnel in unit I at time t.

$\dfrac{\delta m_I}{\delta t}$ = number of men lost in last time step.

$n_{I_1}(0)$ = number of pieces of first equipment on unit I's equipment list at time 0.

$n_{I_1}(t)$ = number of pieces of first equipment on unit I's equipment list at time t.

$\dfrac{\delta n_{I_1}}{\delta t}$ = number of pieces of first equipment on unit I's equipment list in last time step.

After a unit undergoes a change in its operational state, a second pass is made through the IOPTB1 array by subroutine CHGOPN to determine if a second change is required. No more than two operational state changes will be made to a unit in a given time step.

If the value of IOPTB2, representing criteria number, etc., is negative, it is set to positive before extracting the packed data. After processing the data, the corresponding entry in IOPTB1(I,1) is set to 999999999, effectively eliminating that entry from the table for the remainder of the exercise.

Currently, just two entries are input to the table at model initialization. These entries are explained below:

IOPTB1	IOPTB2	DECVAL
01000001300	27001150	.30
00002075137	01002692	5

The first entry causes a unit of type 01 (mech infantry), operational state 13 (attack), (taken from the UU and YY portions of the IOPTB1 doubleword), to evaluate criterion 27 against the data value specified in DECVAL(1), which is .30. Criterion 27 evaluates the fraction of men killed in the unit from model start time. If 30% of the unit's personnel have been killed, the unit should change to operational state 15 (attack stalled).

The second entry causes Blue number 37 with a movement code of 7 (halted), operational state 51 (support fire at band 1), to evaluate criterion number 01, which compares model time to DECVAL(2). If the model time is greater than or equal to 5 minutes, the unit changes its operational state to 69 (artillery non-firing). If it is a member of an operational grouping, it will leave its operational grouping.

4.13 SUPPRESSION CRITERION SELECTION TABLE

The Suppression Criterion Selection Table consists of the following array:

ISSLCT (100)

4.13.1 English-language Description

There are two separate cumulative methods used in CATTS to calculate suppression effects. The suppressive effects of such things as casualty rates are computed each time step (nominally, one minute long) for each simulated unit and stored as a number between zero (not suppressed) and one (completely suppressed). This fraction of men and equipment in the unit will be considered "suppressed" -- that is, nearly invulnerable to fire, unable to move, and unable to return fire -- for the duration of the time step. This "suppression" fraction is calculated at the beginning of each time step, based on occurrences during the previous time step. by subroutine SUPRES. The actual suppression of the men and equipment is effected by a method to be described later. The second method by which suppression effects are included is intended to account for suppressive effects caused by the tactical situation rather than actual casualties. It is accomplished as follows:

In CATTS, men and equipment are always operating in one of eight "modes". These modes are characterized by a rate of fire, a rate of movement, an ammunition type, and a personnel vulnerability class for each equipment type. Each time step, each equipment type in each unit is fractionally distributed over the eight modes, based on the current tactical situation for that unit. The modes, with exceptions which will be explained later, are user-defined by means of the input tables so that a proper input definition of the modes will allow virtually any desired combination of movement, firing, and vulnerability to be achieved, and allow different combinations for each equipment in the unit.

An exception is that mode one is assumed to be a suppressed mode and should always be so defined by the input. In fact, the suppression of the

fraction of the unit calculated by SUPRES is accomplished by placing that fraction of personnel and equipment into mode one before distributing the remainder over the modes. The mode distribution selected for an equipment may place as many additional pieces of equipment into the suppressed mode one as is desired. For example, in a mechanized infantry unit facing an armored unit, it would be possible, and might be desirable, to define the mode inputs in such a way that anti-armor weapons (and the personnel manning them) became very active, while other equipment (and related personnel) became relatively or completely suppressed.

The combination of the two suppressive capabilities provides a comprehensive and realistic suppression modeling capability to the CATTS model. However, since both models are table driven, the ultimate responsibility for tactical realism falls upon the realism of the CATTS data base.

4.13.2 Programmatical Description

The CATTS suppression model contained in subroutine SUPRES calculates the fraction of each unit suppressed at the beginning of each time step and stores the result in the array SIGU for use later. The calculation for a unit for a given time step may use any one of four criteria:

1) Concentration of support fire received during the previous time step (= total number of support fire rounds received, divided by unit area).

2) Rate of personnel loss during previous time step if unit were unsuppressed (= number of personnel who would have been lost if suppression were ignored, divided by the length of the time step).

3) Fraction of original personnel lost per time step during previous time step if unit were unsuppressed (= number of personnel who would have been lost if suppression were ignored, divided by the length of the time step times the number of original personnel).

4) Fraction of original personnel lost since the beginning of the simulation (= original number of personnel minus current number, divided by original number).

The number obtained from the criterion selection table determines which suppression curve is to be used in evaluating the unit's suppression, and

which criterion is to be evaluated and interpolated on the selected curve. The Suppression Curve Table is discussed separately in this section.

ISSLCT (100) is an input table which allows the user to specify which criterion and which curve should be used to calculate the suppression for any given unit. The structure of this table is shown as Figure 4-1 . Essentially, it allows the specification of suppression curve and criterion as a function of unit type (armor, mechanized, etc.), force (red or blue), and operational state (move to contact, mounted attack, area defense, etc.). There is currently space for one hundred entries in this table.

look-up portion	results portion
NNTT TTTS SSSS SSFF	NNNN NNNN NEEE CCC

(Each character represents a bit in the thirty-two bit computer word.)

TTTTT = unit type
SSSSSSS = operational state
FF = red or blue
EEE = criterion to use
CCCC = curve to use
N = unused bits

Figure 4-1 . Structure of Suppression Selection Table ISSLCT

which criterion is to be evaluated and interpolated on the selected curve. The Suppression Curve Table is discussed separately in this section.

ISSLCT (100) is an input table which allows the user to specify which criterion and which curve should be used to calculate the suppression for any given unit. The structure of this table is shown as Figure 4-1. Essentially, it allows the specification of suppression curve and criterion as a function of unit type (armor, mechanized, etc.), force (red or blue), and operational state (move to contact, mounted attack, area defense, etc.). There is currently space for one hundred entries in this table.

look-up portion	results portion
NNTT TTTS SSSS SSFF	NNNN NNNN NEEE CCC

(Each character represents a bit in the thirty-two bit computer word.)

TTTTT = unit type
SSSSSSS = operational state
FF = red or blue
EEE = criterion to use
CCCC = curve to use
N = unused bits

Figure 4-1. Structure of Suppression Selection Table ISSLCT

4.14 RELIEF-VEGETATION ASSOCIATION TABLE

The Relief-Vegetation Association Table is given by the data stored in the FORTRAN integer array IVGLOC(2,64).

4.14.1 English-language Description

The CATTS terrain model is represented by explicit data inputs describing relief, vegetation, and soil. Relief data is organized such that the entire area of operation is partitioned into grid squares. At each grid corner, surface elevation data is known. The surface elevation at points not lying on a grid corner is approximated by interpolation. Specifically, the elevation is determined to be the weighted average of four corner elevations of the grid square in which the point lies. The size of the grid square (called grid resolution) is a user defined input which depends upon the availability of grid elevation data. Presently, the CATTS model uses digitized data having 25.4 meter resolution. Because of the fine resolution, the amount of elevation data required to model relief over the entire area of operation is enormous. This necessitates an I/O scheme to read data quickly and efficiently.

Since all of the relief data will not fit into core storage at once, the entire area of operation is divided into 64 equal blocks. The blocks are ordered and numbered as illustrated below:

Y INDICATOR FOR TERRAIN BLOCK								
8	57	58	59	60	61	62	63	64
7	49	50	51	52	53	54	55	56
6	41	42	43	44	45	46	47	48
5	33	34	35	36	37	38	39	40
4	25	26	27	28	29	30	31	32
3	17	18	19	20	21	22	23	24
2	9	10	11	12	13	14	15	16
1	1	2	3	4	5	6	7	8

X INDICATOR FOR TERRAIN BLOCK

The relief data associated with each block is read into core once and only once per time step. All calculations and processing which require data from a given block are done while the block is in core. Any results, be it intermediate or final, is stored away and referenced as needed by the model at a later time during the time step. To reduce the time involved in reading the relief data, two buffers are used. While the data in one buffer is being processed, the next block of data to be entered is read into the other buffer.

Vegetation is modeled by specifying a dominant class of vegetation which is most typical in the area of operation, and tracing out regions (in the form of triangles, rectangles, or circles) of vegetation which differ from that of the dominant class. Data defining classes of vegetation are input by the user. Similarly, the polygons tracing out the regions of nondominant vegetation are user defined by input. Currently a maximum of 225 polygons may be used to simulate global vegetation. Unfortunately, these polygons are spread throughout the area of operation; they cannot be conveniently placed entirely within a block. Thus some polygons will be split by the division lines which partition the area into blocks. It is possible that one polygon will have portions of its interior lying in four different blocks.

Line of sight calculations are time consuming and require a large amount of core. The Relief-Vegetation Association Table was established to alleviate the above constraints. The line segment between an observer unit and a target unit must be partitioned into subsegments according to the different classes of intervening vegetation. This corresponds to subsegments generated by intersection points of the line segment with all vegetation polygons lying between the observer-target pair. The distance of these subsegments must be known in order to derive probabilistic effects caused by mixed vegetation. Mathematically, this problem reduces to solving sets of simultaneous equations. However, since the locations of the observer and target units may be anywhere in the area of operation, all vegetation polygons defined must be examined for possible intersection with the line between the observer and target. This involves checking all polygons (possibly the maximum of 225) for every line of sight calculation (possibly the maximum of 100); needless to say, this can involve a relatively large amount of processing time.

The Relief-Vegetation Association Table attempts to reduce some of the line of sight processing time. The table is based on two characteristics:

1) Line of sight processing is done in a blockwise fashion because relief data is read into the computer by blocks.

2) Vegetation polygons are defined and ordered such that consecutively numbered polygons are in close proximity within the area of operation.

For each of the 64 terrain blocks, the table contains two numbers:

1) The polygon number of the first vegetation polygon in the block.

2) The total number of polygons in the block.

This information reduces the number of polygons to consider; instead of checking all polygons, only those polygons (or portions of polygons) located within the terrain block currently being processed are examined for intersections. Note that polygons lying between several blocks are considered to be situated in each of them.

When an observer-target pair lies entirely within a terrain block, the line of sight verdict is established immediately. However, inter-block verdicts cannot be determined until all required terrain data blocks have been accessed. Intermediate values must be stored away until the appropriate blocks become available for processing. By reducing the number of polygons to be checked each time a terrain block is processed, the line of sight determinations are performed more expediently. The data in the Relief-Vegetation Table was established to achieve this result.

4.14.2 Programmatical Description

The data comprising the Relief-Vegetation Association Table is stored in the FORTRAN integer array IVGLOC(I,J). Data is retrieved by specifying a value for the index J (an integer between 1 and 64 inclusively) which identifies the terrain data block being processed. The index I references the two words of storage associated with each terrain block. The first word contains a code denoting the number of the first vegetation polygon located in the block, and the second word contains an integer indicating the total number of polygons lying in the block. This information allows the line of sight software to examine a subset of the set of vegetation polygons defined in the model. In particular, the subset is composed only of those polygons which lie partially or entirely

within the terrain block currently being processed. The data in this table is used to keep line of sight processing time at a minimum.

The dimension of the IVGLOC array (2 by 64) allows the area of operation to be divided up to a maximum of 64 blocks. Data is defined by the user and entered into the array by subroutine LOSINP. The XEROX BUFFER IN operation is used to input the data. The values in the array remain constant throughout the simulation. The figure below illustrates the structure and organization of the IVGLOC array.

Block No. / Word No.	J=1	J=2	. . . . . .	J=N	. . . . . .	J=64
I=1	No. of 1st polygon in block 1	No. of 1st polygon in block 2	. . . . . .	No. of 1st polygon in block N	. . . . . .	No. of 1st polygon in block 64
I=2	Total No. of polygons in block 1	Total No. of polygons in block 2	. . . . . .	Total No. of polygons in block N	. . . . . .	Total No. of polygons in block 64

The data in array IVGLOC is referenced only by subroutines LOSINP and LOSCOMP. LOSINP initially zeros out the entire array with a FORTRAN DATA statement, then uses the BUFFER IN operation to enter user defined data. LOSCOMP references elements in the array according to the terrain block currently being processed. In particular, LOSCOMP retrieves data to establish values for the FORTRAN variables VEG1 and NRVP. These values are passed via common block DATAB2 to subroutines MICSOL and LINOBS for line of sight processing.

4.15 TABLE OF UNIT VEGETATION AND SOIL CLASSES

The Table of Unit Vegetation and Soil Classes is represented by the halfword packed array IVGSL(100).

4.15.1 English-language Description

A unit located in the area of operation is situated in a specific class of vegetation and in a specific class of soil. During each time step, a check is made to determine whether the unit has moved since the last time step. If so, the unit may have moved into a different class of vegetation and/or soil. The Line of Sight/Terrain Submodule updates any changes in vegetation or soil class resulting from travel by the unit during the previous time step.

4.15.2 Programmatical Description

The integer array IVGSL contains data indicating what class of vegetation and soil each unit is located within. The array is comprised of one hundred words, one word per unit indexed according to unit number. Each word is data packed with the left half-word containing a code for vegetation class and the right halfword containing a code for soil class. The array format is illustrated below:

IVGSL(1)	VEG. CLASS OF UNIT 1	SOIL CLASS OF UNIT 1
IVGSL(2)	VEG. CLASS OF UNIT 2	SOIL CLASS OF UNIT 2
IVGSL(3)	VEG. CLASS OF UNIT 3	SOIL CLASS OF UNIT 3
.	.	.
.	.	.
.	.	.
IVGSL(I)	VEG. CLASS OF UNIT I	SOIL CLASS OF UNIT I
.	.	.
.	.	.
.	.	.
IVGSL(100)	VEG. CLASS OF UNIT 100	SOIL CLASS OF UNIT 100

In order to reference a unit's vegetation or soil class data, the number of the unit must be specified. This number will point to the appropriate word in the IVGSL array. The soil class code can be obtained merely by masking out the bits comprising the first half of the word (i.e., zero out the 16 high order bits). The vegetation code can be established by masking out the 16 low order bits and conducting a left circular shift of 16 bit positions.

At the beginning of each time step subroutine LOSINP updates the data in the appropriate locations of array IVGSL. LOSINP calls subroutines LOSCOMP and SOIL respectively to determine what classes of vegetation and soil each unit is located in. This is automatically done for all units during the first time step of the simulation in order to fill in the array with initial data. However, for every succeeding time step, this determination is made only for units which have moved during the previous time step.

Knowledge of local soil and vegetation features surrounding a given unit is required when computing movement degradation factors and detection verdicts. In particular, subroutine ADJROM derives fractions used to model the effects of vegetation and soil on equipment movement. Generally the class numbers are used to index into other arrays which contain data representing soil and vegetation. Subroutines RADAR, UASCK, and VISUAL are detection routines which reference class dependent data in the computations it performs. The following is a list of subroutines which use or modify data in the IVGSL array:

ADJROM	INPUT	RADAR	VISUAL
CRDLIC	LOSCOMP	SOIL	
DLOSINP	LOSINP	UASCK	

4.16 MINEFIELD DATA TABLE

The data in arrays MINEDATA(20) and MNEFLDXY(4,3,20) comprise values making up the Minefield Data Table.

4.16.1 English-language Description

Minefields are a type of obstacle (type 3) in the CATTS math model which demand special attention. They are unlike the other types of obstacles modeled because of the following:

1) Minefields are the only area obstacles modeled which may comprise of more than one disjoint piece
2) Minefields are the only obstacles modeled which may inflict damage and casualty.

However, like all other obstacle types, minefields are modeled to impede the progress of units and operational groupings.

A minefield is represented by a rectangle defined by a line segment and a width. The line segment is specified by the XY coordinates of its endpoints, and the width is stipulated by a positive integer number. This line segment, when extended infinitely in both directions, is referred to as the center line. Software in the math model will automatically generate the XY coordinates of each of the four corners of the rectangle representing the minefield.

Gaps between a minefield can be modeled. A maximum of two gaps may be associated with a minefield. In order to simulate gaps, the line segment defining the minefield must be divided into disjoint subsegments. These subsegments must lie along the center line of the minefield. Note that separating the line segment into two partitions creates a single gap within the minefield. Similarly, partitioning the line segment into three pieces allows two gaps to be modeled. Again, each subsegment is uniquely defined by the XY coordinates of its endpoints. This data, along with a designated width, permits the software to automatically generate the corners of the rectangles corresponding to each subsegment. Rectangles generated from such subsegments are called sections. In effect, the modeling of gaps makes a minefield appear to be a set of disjoint rectangles (called sections) each having a common width and center line.

Thus far, minefields are the only obstacle type modeled with the capability of simulating damage and casualty. When a unit is detained by a minefield, personnel casualties and/or damage to equipment must be accounted for. This attrition depends mainly upon whether the unit is mounted or dismounted. Dismounted units suffer one personnel casualty when it encounters a minefield. A mounted unit will have the first self-propelled vehicle in its equipment list destroyed; the unit also suffers the expected number of personnel casualties associated with the destruction of this vehicle. Personnel killed are assumed to come from the most vulnerable classes. Casualty and damage statistics are stored into memory for updating and alert generation purposes.

Breaching a minefield consists of a series of calculations that determine the shortest route through the minefield. Since minefields have a rectangle geometry, the shortest path across is usually the path normal to the side of the rectangle containing the point of obstruction. However, the normal path is not necessarily the shortest path. For instance, when a unit's path of travel is such that it cuts the corner of the rectangle representing a minefield, the actual distance to be traversed across may be less that the normal distance. Thus when breaching a minefield in the CATTS model, the path of least distance is established and taken after the unit has suffered delay and casualties.

4.16.2 Programmatical Description

The data making up the Minefield Data Table reside in halfword packed arrays MINEDATA(K) and MNEFLDXY(I,J,K). Data pertaining to a specific minefield is retrieved by stipulating the minefield number K, where K may range in value from 1 to 20 inclusively. Thus the size limitation of the above arrays allows for a maximum of twenty minefields to exist in the model simultaneously. Since gaps can be modeled for a given minefield, the coordinate values of the corners of each section of the minefield can be referenced by identifying the section number J, where J may range in value from 1 to 3 inclusively, and the corner number I, where I may range in value from 1 to 4 inclusively. Note that no minefield may be comprised of more than three sections (hence no more than two gaps may be modeled for a given minefield).

The data stored into the above arrays are not input but are computed from obstacle input data. The obstacle input data exist in minimal form. All that is provided is the width and a set of XY coordinates designating the endpoints of linear subsegments lying along the center line of the minefield; the number of subsegments is also an input quantity. Subroutine MINEFLDS examines the obstacle-type array IOBSTYPE, searching for an obstacle of type three. Type three obstacles are minefields, and, when one is found, data is generated and entered into the arrays MINEDATA and MNEFLDXY. The data in these arrays uniquely identify and represent each minefield in the model.

Subroutine MINEFLDS computes the coordinate values describing each rectangular section of the minefield. In addition, the number of sections comprising the minefield, as well as the obstacle number associated with the minefield, are determined. These data values are entered into the above arrays in the following fashion: for the N-th minefield, the left halfword of the array element MINEDATA(N) will contain the obstacle number (an integer between 1 and 50 inclusively) associated with this minefield, and the right halfword will contain the total number of sections (1, 2, or 3) comprising this minefield. Assuming that the N-th minefield is composed of three sections, the XY coordinates of the L-th corner ($1 \leq L \leq 4$) in the M-th section ($1 \leq M \leq 3$) is stored in the array element MNEFLDXY(L,M,N). Because of the magnitude of the coordinate values relative to word size, the X and Y values each must be rounded off, converted from real number representation to integer number representation, and divided by a factor of four. This transformation is accomplished in order to pack the XY data into halfwords. The data structure and organization of the two arrays comprising the Minefield Data Table is revealed in Table 4-5.

Subroutine MINEFLDS is called only once by FORMAIN, during model initialization, to create the data values. Once the data has been established, it is never modified by subsequent processing; it remains constant throughout the simulation. Caution should be extended when referencing the XY data in array MNEFLDXY. When the respective values are unpacked, they must be multiplied by a factor of four to obtain the true XY values.

The minefield arrays are referenced by the movement logic (in particular, the Obstacle Submodule) of the CATTS math model. Unit interactions with minefields involve damage and casualty assessments, delays, and establishing breach paths. The following subroutines model the above interactions:

BRCHPATH*

OBSDELAY

OBSTACLE*

OBSUPDAT

OBSWIDTH*

(Note: * indicates the subroutine accesses data from minefield arrays.)

Table 4-5 . Minefield Data Table

MINEDATA ARRAY		MNEFLDXY ARRAY											
		1ST SECTION				2ND SECTION				3RD SECTION			
Obstacle No.of 1 Minefield	Number of Sections	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER
		X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER	X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER	X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER
. . .		. . .		. . .		. . .		. . .		. . .		. . .	
Obstacle No. Nth Minefield	Number of Sections	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER
		X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER	X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER	X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER
. . .		. . .		. . .		. . .		. . .		. . .		. . .	
Obstacle No.20th Minefield	Number of Sections	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER	X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 3 CORNER	Y COORD 3 CORNER
		X COORD 1 CORNER	Y COORD 1 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER	X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER	X COORD 2 CORNER	Y COORD 2 CORNER	X COORD 4 CORNER	Y COORD 4 CORNER
HALF WORD		FULL WORD		HALF WORD									

4.17 BREAK RANGE TABLE

The Break Range Table consists of the following array:

MURTAB(150,5)

4.17.1 English-language Description

The Break Range Table contains data enabling the final selection of a mode distribution vector for processing a particular equipment in a single unit firing against a single target unit. The table contains two "break" ranges, which divide range between firing unit and target unit into three categories:

- less than the first break range.
- between the first and second break range.
- greater than the second break range.

Each of the three categories has a specific number associated with it. This number indicates the mode distribution vector to be extracted from the Mode Distribution Vector Table and used for the equipment being processed.

The appropriate entry in the Break Range Table is determined by finding a match on the code words in the Mode Selection Code Table, discussed elsewhere in this section.

All non-firing, direct fire, and indirect fire equipment processed in any minute of simulated time uses the Break Range Table as part of the Ground Fire Module processing. First, the Mode Selection Table is used to determine the appropriate index, then the Break Range Table, using that index, provides two "break" ranges and three mode distribution vector numbers. The model determines which of the three mode distribution vector numbers is appropriate as a function of range to the target unit being fired at. If an equipment is not a weapon, the

procedure is similar to that for firing weapons, but all pieces of the equipment in the unit are distributed among the various modes by one of the three mode distribution vectors associated with mode selection code that fits the case. This code will omit any reference to target unit state or type (these positions in the code word will each be filled by 99 or 00). Again, the final selection among the three vectors associated with the code will depend on the calculated mode selection range. If the equipment is a direct- or indirect-fire weapon, but there are no target units available for it to fire at, then it is treated in exactly the same manner as a nonweapon. Table 4-6 defines the mode selection range for various conditions.

Data is input to the table at model initialization and not altered during an exercise.

4.17.2 Programmatical Description

The MURTAB(I,J) array contains up to 150 five-item rows of data. A single row consists of the following data:

Break Range #1	Break Range #2	Mode Vector #1	Mode Vector #2	Mode Vector #3

The row-index, I, to be used for processing a given piece of ground equipment (in a firing unit against a target unit) is determined by a match in the Mode Selection Table. All five items in the row, $1 \leq J \leq 5$, are required to determine the mode distribution vector to be used in allocating the equipment over the eight modes defined in the model. The following example illustrates the use of the Break Range Table.

Given

A unit is moving towards an opposing unit, is within firing range of his own tanks, but is not engaged with any opposing unit. At time t, the range between the two units is 2500 meters. The Mode Selection Table, MUSLCT, indicates that the first entry in the Break Range Table is to be

used to determine the mode distribution vector number. In the FEBA Gold scenario, MURTAB(1,J) for J = 1, 2, 3, 4, 5, respectively, is:

Break Range #1	Break Range #2	Mode Vector #1	Mode Vector #2	Mode Vector #3
2000	2000	13	11	11

Processing

Subroutine SCHMU is called by ORGFIR to process MURTAB after SCHRMU has found the matching code word and appropriate index. The mode selection range, MUSLRA, is the distance between units (see Table 4-6). SCHMU compares the range of 2500 meters with both the first and second break ranges. Since it exceeds the second range, the mode vector #3 entry of 11 is the desired mode distribution vector for the weapons allocated against that target unit.

If, as the exercise progresses, the firing unit has moved to within 1500 meters at time t + 5, mode distribution vector 13 will be used, since the mode selection range is less than the first break range.

It should be noted that all rows MURTAB in the current scenarios have both break ranges set to the same value, so that for a given row, at most two different modes are defined.

Also, only 44 of the 150 possible rows in the array are used in these scenarios.

Table 4-6 . Determination of Mode Selection Range

OPERATING CONDITIONS OF THE FIRING UNIT	MODE SELECTION RANGE, MUSLRA
The firing unit is engaged and exchanging direct- or indirect-fire with enemy units.	MUSLRA = distance to enemy FEBA <u>or</u> distance to nearest enemy unit at which I may fire, whichever is less.
The firing unit is not engaged but is exchanging direct- or indirect-fire with enemy units.	MUSLRA = distance to nearest enemy unit at which I may fire.
The firing unit is engaged but is not within range to fire at enemy units.	MUSLRA = distance to enemy FEBA.
The firing unit is not engaged and is not within range to fire at enemy units.	MUSLRA is artificially large to ensure selection of the third vector number.

4.18 MODE SELECTION CODE TABLE

The Mode Selection Code Table consists of the following array:

MUSLCT(150)

4.18.1 English-language Description

The Mode Selection Table contains code words enabling a unit about to fire at an opposing unit to determine which entry in the Break Range Table to use as a function of the

- firing unit's operational state.
- target unit's operational state.
- target unit's type.
- equipment type being distributed in the firing unit.

Having found the appropriate match, the corresponding position in the Break Range Table is used to determine the mode distribution vector for the equipment.

All ground equipment types use the Mode Selection and Break Range Tables to determine their mode distribution vector except support fire weapons, which use the Fire Support Table, and air defense weapons, which use a single mode distribution. Therefore, non-firing equipments (e.g., trucks), direct fire weapons (e.g., rifles), and indirect fire weapons (e.g., light mortars) all use the Mode Selection Table. These equipment type are processed as part of the Ground Fire Module.

Data is loaded into the table during initialization of the model, and not changed throughout an exercise. The present Mode Selection Table code entries in the three scenarios, FEBA Gold, FEBA Silver, and Attack require a match on just one of the four variables listed above, that is, the firing unit's operationaL state. The other three variables are ignored. Simple re-definition of the table entries could utilize any or all of the other three capabilities.

4.18.2 Programmatical Description

The Mode Selection Table is input at model initialization time by subroutine MUINP. Thereafter, it is referenced only by subroutine SCHRMU, which is only called by subroutine ORGFIR. SCHRMU is a service routine whose sole function is to find a match for the appropriate

- fire unit operational state.
- target unit operational state.
- target unit type
- weapon type.

ORGFIR and, subsequently, SCHRMU are called by three subroutines in the Ground Fire Module for three different categories of equipment:

Subroutine	Equipment Category
NOTGT	non-firing equipment
FIRALO	direct and indirect fire weapons against a target
WTSUB	direct and indirect fire weapons having no targets

The processing of every non-firing, direct, and indirect firing equipment in every unit requires a search of this table to determine the proper mode distribution vector for the equipment.

An entry in the MUSLCT array consists of 8-digits of data, of the form WWXXYYZZ, where

WW = operational state of firing unit (00 => any state. If WW is a multiple of 10 (not 00), it represents all states having the same first digit).

XX = operational state of target unit (00 => any state. If XX is a multiple of 10 (not 00), it represents all states having the same first digit. 99 => there is no target unit).

YY = target unit type (00 => any type; 99 => there is no target unit).

ZZ = equipment type (00 => any type).

Subroutine SCHRMU checks each entry by integer-dividing by 1000000 to compare the "WW" portion of the entry against the present firing unit. If this check is passed, the remainder of the above division is successively integer-divided by 10000 and 100 to yield XX and YY. The remainder of the final division is ZZ. An additional check is made for the special codes 00 and 99. If the four codes compare favorably with the entry, the index is saved to extract the corresponding values in the Break Range Table.

It is not always necessary to provide an entirely new set of mode selection codes for a unit's equipment every time it changes operational state or every time its target unit changes state. The program allows for the use of the same mode selection codes for blocks of firing unit states and blocks of target unit states. This is done by indicating a unit's state in the code as a multiple of 10. For example, if the state of the firing unit is given in the code as 40, this code applies also to firing units in states 41, 42, ..., 49. The same rule holds for target units. If, however, the state of the firing unit is given in the code as 42, then this code would apply to that state only.

The MUSLCT array holds a maximum of 150 entries. Presently, just 44 entries are defined by the three existing scenarios, and these entries have only firing unit operational states defined. Therefore, once a match is made on firing unit operational state with the present scenarios, the index into the Break Range Table is completely determined.

4.19 MOVEMENT CODE CHANGE TABLE

The Movement Code Change Table is represented by data contained in the following arrays:

MVCHG1(90,2), MVCHG2(90), MVDATA(100).

4.19.1 English-language Description

The manner of movement by a unit varies continuously as situations and conditions arise throughout the simulation. The Movement Code Change Table provides the capability of transitioning from one manner of movement to another without having to exercise (interactively) the maneuver command and control menu. Recall that at any point of time during the simulation, the movement code identifies which of sixteen different ways a unit is moving. Movement data accompanies each code to facilitate the desired manner of movement. This table is used to sequence in a logical and orderly fashion the movement codes so that a unit can perform required maneuvers to achieve objectives.

Initially, movement is specified by user defined inputs. To modify the manner of movement, a new movement code along with its appropriate data must be established. The modification can be implemented in two ways: (1) by built-in software in the math model (i.e., hard-coded logic), or (2) by a table look-up scheme (i.e, Movement Code Change Table). Examples of the former are triggered by the following occurrences:

1) Maneuver command and control (interactive)
2) Enemy encounter (engagement)
3) Obstacle encounter.

When a controller establishes a maneuver command and control event notice, the Maneuver Events Processor is the software module which updates all movement data base variables to effect the desired change in movement. All unit movement during an engagement is controlled by logic rules and decisions built into the Engagements Module. Similarly, the changes in movement data required for a unit to negotiate its way through an obstacle is hard-coded into the Obstacle Submodule. Note that in the three instances above, movement modification is accomplished by rules and decisions implanted directly in the software; these rules are fundamental and remain invariant to data

base changes. Specifically these rules are established independent of data inputs and cannot be changed unless the software (math model) is redesigned.

The latter approach, using a table look-up scheme, provides the capability of establishing movement change rules through user defined inputs. These rules have the advantage of changeability by the user, without impacting the design of the math model. Specifically, the user is given the opportunity to vary the decision logic used to modify movement data between simulation exercises. The table look-up procedure is automatically invoked whenever the following circumstances transpire:

4) Arrival at destination
5) Change of operational state.

The above occasions activate a search of the Movement Change Code Table. The search is based upon several unit characteristics:

1) Unit type
2) Operational state
3) Red or blue army
4) Current movement code
5) Operational group number
6) Unit number

The table is organized such that new movement data may be retrieved for a unit, whenever the above six considerations indicate a match with the new data's associated index words. New data for the unit consists of the following:

1) New movement code
2) New movement data associated with the new code
3) New deployment code
4) New operational state (if applicable).

Note that if the search does not produce a match, the unit's movement data will not change; its present movement status is retained.

The Movement Code Change Table provides for a maximum of ninety different change rules to affect unit movement. These rules are accessed whenever a unit either changes operational states or arrives at a destination. Each rule can be thought of as consisting of two parts: (1) the index portion and (2) the new data portion. The index portion is comprised of two computer words

packed with codes stipulating which characteristics must be satisfied by a unit in order to assume the new data given by the rule. The format of the new data portion is illustrated in Figure 4.19-1. When a match occurs, the unit acquires the new data given by the rule. Additional data related to the new movement is also calculated for the unit. Note that a match does not guarantee that a new movement status may be assumed. One very important restriction on the use of the table is the following: a unit may not be changed from an unengaged movement code (5-16) to an engaged movement code (1-4). Such a change is always performed automatically by the built-in engagement decision logic. The table may, however, change a unit's movement code from an engaged code to an unengaged code when it is far enough from the enemy FEBA to be considered out of range. Another restriction on the table is that any attempt to change a unit's movement code to 4 (deployed and waiting when engaged) or 16 (deployed and waiting when unengaged) is ignored. This function is already handled by built-in arrival logic in the Movement Module. The Movement Code Change Table provides the capability of automatically transitioning the movement status of a unit when it arrives at its destination or undergoes a change in operational state.

A table look-up scheme referencing decision rules used to implement movement changes has several advantages. As mentioned before, the decision rules can be changed by user input, thereby avoiding redesign of math model software. The Movement Code Change Table is established at model initialization time and remains constant throughout the simulation. Another advantage lies in the model being provided with the capability of instructing units to assume new movement directives without requiring the controllers to accomplish this interactively. This frees the controller from a time consuming task. Advantage can also be seen in the realism achieved when a unit can be made to change its manner of movement as a function of six different factors currently attributed to the unit during the simulation.

It is worth noting that in the absence of movement changes being sequenced by the engagement or obstacle logic or by the Maneuver Events Processor, a unit's movement is determined by the Movement Code Change Table. At any given time during the simulation, unit movement is controlled by one and only one of the following types of sequencing:

1) Interactive Maneuver Command and Control

2) Built-in software decision logic (engagement, obstacle)

3) Movement Code Change Table.

There exist a hierarchy of control wherein each type of sequencing takes precedence over any type of sequencing below it. For instance, sequencing by the table method can be overriden by sequencing done by both interactive control and built-in decision software. Movement changes accomplished by built-in logic can be supplanted by interactive commands but not by the table rules. This of course means that the interactive capability receives highest priority in transitioning movement status. The Movement Code Change Table, therefore, provides the nominal method of sequencing unit movement changes.

4.19.2 Programmatical Description

The Movement Code Change Table is composed of data found in the following arrays: MVCHG1(I,J), MVCHG2(I), MVDATA(K). The data in these arrays provide decision rules for units to change their manner of movement. Each entry in the table is a rule consisting of two parts. The first part given by data in the array MVCHG1 serves as an index which will direct the selection of new movement data for a given unit. Each index is two words (J = 1,2) in length and packed with specific codes which dictate the conditions and characteristics that a unit must meet before it is permitted to assume the new movement data provided. Associated with each two word index is the second part of the rule: new movement data packed into a single word residing in the array MVCHG2. For each index specified by MVCHG1, a corresponding element in MVCHG2 will contain new data to transition the manner of unit movement (provided the unit satisfies all criteria given in the two word index of MVCHG1). Additional data values may be required to facilitate a new move. This additional data is stored in the integer array MVDATA ($1 \leq K \leq 100$); it is referenced by a code packed along with new data codes in the associated word of MVCHG2. Currently a maximum of ninety such decision rules may be defined; each is referenced by subscript I of arrays MVCHG1 and MVCHG2 ($1 \leq I \leq 90$) respectively. Note that MVCHG1 requires a pair of words to specify an index entry. Each word of the pair is referenced by subscript J ($1 \leq J \leq 2$).

The decision rules are established by user-defined inputs. They are read into the arrays by subroutine MOVINP during model initialization and remain constant throughout the simulation. The formats of packed data residing in arrays MVCHG1 and MVCHG2, respectively, are given in Figure 4-2 below:

MVCHG1 (I,J) = I^{th} entry in movement code change table for changes in movement code of units. Each entry is of the form UVVWXXYY, where
0 in any position means equality of look up.
If YY = 99, unit can be in an operational grouping
MVCHG1 (I,1) = UUVVW
MVCHG1 (I,2) = XXYYZZ
(1 < = I < = 90)
Such that
UU = Unit type
VV = Operational state of unit
W = 1 or 2 (red or blue)
XX = Current movement code
YY = Operational grouping
ZZ = Unit number.
Equality of look up means use MVCHG2(I) to find new data for unit.

MVCHG2 (I) = This table has an entry corresponding to each entry I in the movement code change table. Table MVCHG1 entries in MVCHG2 give new unit data and are of the form EEFGJJKKK, where
MVGCHG2(I) = EEFGJJKKK
Such that
EF = New movement code of unit
F = New travel code
G = New deployment code (0 => not used)
JJ = New operational state (0 = no change), new state can't be used here for units that have just had a change in op state via the change of state table.
KKKK = Line number in movement data table (MVDATA) associated with new movement code.

MVDATA (K) = Movement data value referred to by KKK code of MVCHG2.

Figure 4-2 . Movement Change Code Table Formats

Note: These rules can be changed between simulation exercises, merely by redefining the data.

Data in the arrays comprising the Movement Code Change Table is accessed by subroutine NEWMOV. Whenever a unit arrives at a destination or has its state changed by satisfying one of the change-of-state criteria, subroutine NEWMOV is called to determine whether or not the unit's movement code is to be changed and, if so, to what value. The information describing the characteristics of the unit is compared against the information packed in the two word index entries of array MVCHG1. The following factors form the basis of comparison:

1) Unit type
2) Operational state
3) Red or blue army
4) Current movement code
5) Operational group number
6) Unit number.

If the data in one of the index entries of MVCHG1 matches the data describing the unit's characteristic, the associated new movement word given in array MVCHG2 is referenced. This word contains packed data codes used to change the unit's movement status. In particular, the following codes may be modified:

1) Movement code
2) Travel code
3) Deployment code
4) Operational state (if applicable).

If additional movement data must be supplied to facilitate the movement change, array MVDATA must be accessed. This data is retrived by referencing the line number code given in the associated new data word of MVCHG2.

When no match occurs, the unit will retain its current movement status. Note that finding a match among the entries of array MVCHG1 does not guarantee a movement change. One very important restriction is the following: units may not be changed from an unengaged (5-16) to an engaged (1-4) movement code. Such a change is performed automatically by built-in software during arrival and engagement processing. The same is true when a unit's movement code is being changed to 4 or 16 (deployed and waiting). Built-in software will

handle this, and any attempt to produce such a change (via arrays MVCHG1 and MVCHG2) is ignored.

The structure and organization of the Movement Code Change Table is illustrated in Table 4-7 .

Table 4-7 . The Arrays Representing the Movement Change Code Table

INDEX ENTRIES ARRAY

ENTRY NO. 1	WORD 1	WORD 2
1	MVCHG1(1,1)	MVCHG1(1,2)
2	MVCHG1(2,1)	MVCHG1(2,2)
3	MVCHG1(3,1)	MVCHG1(3,2)
⋮	⋮	⋮
N	MVCHG1(N,1) UUVVW	MVCHG(N,2) XXYYZZ
⋮	⋮	⋮
90	MVCHG1(90,1)	MVCHG1(90,2)

N-th Entry

ASSOCIATED NEW DATA ARRAY

ARRAY MVCHG2
MVCHG2(1)
MVCHG2(2)
MVCHG2(3)
⋮
MUCHG2(N) EEFGJJKKK
⋮
MVCHG2(90)

KKK-th Entry

ASSOCIATED MOVEMENT DATA ARRAY

ARRAY MVDATA
MVDATA(1)
MVDATA(2)
MVDATA(3)
⋮
MVDATA(KKK)
⋮
MVDATA(100)

Note: The codes given by UUVVW, XXYYZZ, and EEFGJJKKK are defined in Figure 4.19-1.

4.20 MOVEMENT DATA TABLE FOR OPERATIONAL GROUPINGS

The Movement Data Table For Operational Groupings consist of the following arrays:

MTCDOG(20), MVDTA1(20), MVDTA2(20), MVDTA3(20)

4.20.1 English-language Description

The manner in which operational groupings move can be described entirely by data contained in the Movement Data Table For Operational Groupings. For each operational grouping, a movement code, along with up to three data values, specify its movement. Movement code is an integer (1 through 16) which distinguishes the sixteen different ways an operational grouping may move. The associated data values present details which facilitate the desired manner of movement.

As long as an operational grouping is active, it must have a valid movement code. The code is initialized by user defined inputs, and is constantly modified thereafter. Normally, a sequence of movement codes is achieved as various situations arise during the simulation. Specifically, the movement code is changed whenever the operational grouping arrives at its destination or encounters the enemy. In addition, the movement code is manipulated interactively by the use of the maneuver control menu.

The types of movement achievable by an operational grouping can be distinguished by two classes. One class describes the manner of movement when the operational grouping is engaged, whereas the other is concerned with unengaged maneuvers. The first class includes movement codes 1 through 4, which describes the following respectively:

1. automatic movement when confronting the enemy in an engagement
2. withdrawing from an engagement
3. in the process of deploying in an engagement
4. deployed in an engagement, waiting for others to deploy

In all instances, the integer identifying the engagement is an associated data value accompanying each of the engaged movement codes. The deploying codes require, in addition to the engagement number, the X and Y coordinate values specifying the deployment location.

The other class includes movement codes 5 through 16. These codes control maneuvers when the operational grouping is not in an engagement. They present a variety of ways for an operational grouping to move. This includes no movement

(halted), movement in a fixed direction, movement along a predefined route, movement involved with deployment, and movement toward a specific point. Note that a specific point may consist of the following:

1. a fixed XY location
2. a point relative to another operational grouping (friendly or enemy)
3. a point relative to a friendly or enemy engagement FEBA
4. a point relative to a unit (friendly or enemy)

To implement the desired manner of movement, specific associated data values must be given along with the movement code. This is illustrated in Table 4-8 . Please note that the codes and data values used to specify movement of operational groupings are exactly the same as those used to dictate movement of units.

4.20.2 Programmatical Description

The Movement Data Table For Operational Groupings is comprised of data found in the following integer arrays: MTCDOG(I), MVDTA1(I), MVDTA2(I), and MVDTA3(I). Data is retrieved by specifying the integer identifying the operational grouping. This is a number between 1 and 20 inclusively. Note also that the size limitations of the above arrays imply that movement data can be defined for at most 20 operational groupings.

The array MTCDOG contains the movement code of each operational grouping. This code is an integer between 1 and 16 inclusively (see Table 4-8). Arrays MVDTA1, MVDTA2, and MVDTA3 contain, for each operational grouping, associated data values required to achieve the desired manner of movement. Depending upon the movement code, data found in these arrays have assorted meanings. This includes values defining engagements, units, operational groupings, destination (i.e. XY) coordinates, direction in degrees, routes, route points, etc. Table 4-8 gives all details relating movement codes and associated data values.

The arrays comprising the Movement Data Table For Operational Groupings are dynamic. This means that data is entered initially by pre-defined input, but is changed according to situations (notably arrival at designated points or encounters with the enemy) which arise during the simulation. Furthermore, the data can be manipulated interactively by maneuver command and control. Table 4-9 illustrates the structure and organization of the arrays.

Table 4-8. Movement of Units and Operational Grouping

Movement Code (MVTCD or MTCDOG)	Description of Movement	First Movement Data Value (MVDT 1 or MVDTA 1)	Second Movement Data Value (MVDT 2 or MVDTA 2)	Third Movement Data Value (MVDT 3 or MVDTA 3)
Engaged[1/]				
1	Normally engaged [2/]	Engagement number	-	-
2	Withdrawing from engagement	Engagement number	-	-
3	Deploying in an engagement[3/]	Engagement number	Destination coordinate, x[4/]	Destination coordinate, y[4/]
4	Deployed in an engagement, waiting for others to deploy[5/]	Engagement number	Location coordinate, x[4/]	Location coordinate, y[4/]
Unengaged				
5	Moving in a fixed direction	(Direction in degrees)	-	
6	Moving along a route	Number of next point on route curve	Route curve number	Route or control measure flag =1 unit is following a route (IXPTH, IYPTH), =2 unit is following a control measure
7	Halted	(If unit or operational group is defunct, former engagement number or 0 is given)		

Table 4-8 . Movement of Units and Operational Grouping (Continued)

Movement Code (MVTCD or MTCDOG) Unengaged	Description of Movement	First Movement Data Value (MVDT 1 or MVDTA 1)	Second Movement Data Value (MVDT 2 or MVDTA 2)	Third Movement Data Value (MVDT 3 or MVDTA 3)
8	Moving toward specific fixed point	Destination coordinate, x	Destination coordinate, y	-
9	Moving toward point relative to a friendly operational group 6/	Destination operational group number	Destination coordinate relative to operational group, forward direction, $\delta\varepsilon$	Destination coordinate relative to operational group, lateral (+ left) $\delta\eta$
10	Moving toward point relative to an enemy operational group 7/	Destination operational group number	Destination coordinate relative to operational group, forward direction, $\delta\varepsilon$	Destination coordinate relative to operational group, lateral (+ left), $\delta\eta$
11	Moving toward point relative to a friendly engagement FEBA	Number of a friendly operational group in the destination engagement 8/	Destination coordinate relative to FEBA, forward direction, $\delta\varepsilon$	Destination coordinate relative to FEBA, lateral (+ left), $\delta\eta$
12	Moving toward point relative to an enemy engagement FEBA 9/	Number of an enemy operational group in the destination engagement 8/	Destination coordinate relative to FEBA, forward direction, $\delta\varepsilon$	Destination coordinate relative to FEBA, lateral (+ left), $\delta\eta$

Table 4-8. Movement of Units and Operational Grouping (Continued)

Movement Code (MVTCD or MTCDOG) Unengaged	Description of Movement	First Movement Data Value (MVDT 1 or MVDTA 1)	Second Movement Data Value (MVDT 2 or MVDTA 2)	Third Movement Data Value (MVDT 3 or MVDTA 3)
13	Moving toward point relative to a friendly unit[10/]	Destination unit number	Destination coordinate relative to unit, forward direction, $\delta\varepsilon$	Destination coordinate relative to unit, lateral (+ left), δn
14	Moving toward point relative to an enemy unit[7/]	Destination unit number	Destination coordinate relative to unit, forward direction, $\delta\varepsilon$	Destination coordinate relative to unit, lateral (+ left), δn
15	Deploying, not engaged	(Direction that operational group will face when deployed)[12/]	Destination coordinate, x[4/]	Destination coordinate, y[4/]
16	Deployed, not engaged, waiting for other to deploy[11/]	-	Location coordiante, x[4/]	Location coordinate, y[4/]

1/ Units change from an unengaged movement code to an engaged code only by automatic means; i.e., they cannot be directed to make such a change.

2/ Units not in an operational grouping have this movement code assigned automatically when they become engaged.

3/ Units belonging to an operational grouping have this code assigned automatically when they become engaged. If a deployed operational grouping is not desired, units must be changed to movement code 1 by changing their state.

Table 4-8 . Movement of Units and Operational Grouping (Continued)

4/ Determined automatically by the computer.

5/ As each unit of an operational grouping reaches its deployed location, its movement code is automatically changed to 4 unless specifically directed to another value. When all units of an operational grouping have deployed, the movement code of each is changed automatically to 1 unless the movement code of the operational grouping has been changed to some value other than 3 or 4 in the meantime.

6/ If the operational grouping is engaged, the arriving unit will join the same engagement.

7/ If the operational grouping or unit is engaged, the arriving unit will join this engagement or start a new one, depending on its position relative to the friendly FEBA. If the operational grouping or unit is not engaged, the arriving unit will start one.

8/ If the destination operational grouping is not engaged, the operational grouping's coordinates are used in place of the FEBA coordinates.

9/ Unit does not engage upon arrival.

10/ If the destination unit is in an operational grouping, the arriving unit will join the same operational grouping.

11/ As each unit reaches its deployed location, its movement code is automatically changed to 16 unless specifically directed to another value. When all units of an operational grouping have deployed, the movement code of each is changed automatically to 7 unless the movement code of the operational grouping has been changed to some value other than 15 or 16 in the meantime.

12/ This direction must be entered in the Movement Data Table (MVDATA), and its location in the table noted in code work MVCHG2. If, however, the new movement code of the operational grouping is not specified, the direction used will be the current direction of movement of the operational grouping. The value is not actually put into MVDTI or MVDTAI.

Table 4-9 . Arrays Comprising Movement Data Table for Operational Groupings

Operational Grouping No.	MTCDOG	MVDTA1	MVDTA2	MVDTA3
1	Movement Code 1st Opr Group	1st associated data value-1st Opr Group	2nd associated data value-1st Opr Group	3rd associated data value-1st Opr Group
2	Movement Code 2nd Opr Group	1st associated data value-2nd Opr Group	2nd associated data value-2nd Opr Group	3rd associated data value-2nd Opr Group
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
20	Movement Code 20th Opr Group	1st associated data value-20th Opr Grp	2nd associated data value-20th Opr Grp	3rd associated data value-20th Opr Grp

4.21 MOVEMENT DATA TABLE FOR UNITS

The Movement Data Table For Units consist of the following arrays:

MVTCD(100), MVDT1(100), MVDT2(100), MVDT3(100)

4.21.1 English-language Description

The manner in which units move can be described fully by data residing in the Movement Data Table For Units. Every active unit has a movement code, along with up to three associated data values, to specify its movement. Movement code is an integer (1 through 16) which distinguishes the sixteen different ways a unit may move. The associated data values present details which facilitate the desired manner of movement.

All non-defunct units must possess a valid movement code. The code is initialized by user defined inputs, and is constantly modified thereafter. Normally, a sequence of movement codes is achieved as various situations arise during the simulation. Specifically, the movement code is changed whenever the unit arrives at its destination or encounters the enemy. Furthermore, the movement code can be modified interactively through the use of the maneuver control menu.

The kinds of movement attainable by a unit can be separated into two classes. One class pertains to the manner of movement associated with a unit when it is engaged, whereas the other deals with movement when the unit is unengaged. The former includes movement codes 1 through 4, which described the following respectively:

1. automatic movement when confronting the enemy in an engagement
2. withdrawing from an engagement
3. in the process of deploying in an engagement
4. deployed in an engagement, waiting for others to deploy

In all instances, the integer identifying the engagement is an associated data value accompanying each of the engaged movement codes. The deploying codes require, in addition to the engagement number, the X and Y coordinate values specifying the deployment location.

The latter refers to movement codes 5 through 16. These codes control maneuvers when the unit is not in an engagement. A variety of ways is provided for unengaged units to move. This includes no movement (halted), movement in a fixed direction, movement along a pre-defined route, movement involved

with deployment, and movement toward a specific point. Note that a specific point may consist of the following:

1. a fixed XY location
2. a point relative to another operational grouping (friendly or enemy)
3. a point relative to a friendly or enemy engagement FEBA
4. a point relative to a unit (friendly or enemy)

To implement the desired manner of movement, specific associated data values must be given along with the movement code. This is illustrated in Table 4-8. Note that the codes and data values used to specify the movement of units are exactly the same as those used to dictate movement of operational groupings.

4.21.2 Programmatical Description

The Movement Data Table For Units is comprised of data contained in the following arrays: MVTCD(I), MVDT1(I), MVDT2(I), and MVDT3(I). Data is referenced by stipulating the integer identifying the unit. This is a number ranging in value from 1 to 100 inclusively. The size limitations of the above arrays allow movement data to be defined for no more than 100 units.

The array MVTCD contains the movement code for each unit. This code is an integer between 1 and 16 inclusively (see table 5.5-1). Arrays MVDT1, MVDT2, and MVDT3 contain, for each unit, associated data values required to achieve the desired manner of movement. Depending upon the movement code, data residing in these arrays have assorted meanings. This includes values defining engagements, units, operational groupings, destination (i.e. XY) coordinates, directions in degrees, routes, route points, etc. Table 4-8 presents all details relating movement codes and associated data values.

The arrays making up the Movement Data Table For Units are dynamic. This means that data is entered initially by pre-defined input, but is modified according to situations (notably arrival at designated points or encounters with the enemy) which arise during the simulation. Furthermore, the movement data can be manipulated interactively by maneuver command and control. Table 4-10 illustrates the structure and organization of the arrays.

Table 4-10. Arrays Comprising Movement Data Table for Units

Unit Number	MVTCD	MVDT1	MVDT2	MVDT3
1	Movement Code for unit 1	1st associated data value for unit 1	2nd associated data value for unit 1	3rd associated data value for unit 1
2	Movement Code for unit 2	1st associated data value for unit 2	2nd associated data value for unit 2	3rd associated data value for unit 2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
100	Movement Code for unit 100	1st associated data value for unit 100	2nd associated data value for unit 100	3rd associated data value for unit 100

4.22 VEGETATION CLASS DATA TABLE

The Vegetation Class Data Table contains data used to represent the terrain feature of vegetation; the data is stored in the FORTRAN arrays:

H(16,4), W(16,4), RHO(16,4).

4.22.1 English-language Description

The CATTS math model simulates sixteen different classes of vegetation. Each class of vegetation consists of features of different types. A feature has three attributes: height, width, and density. By features we mean objects each having the geometry of a right circular cylindrical solid. Within a region, these vegetation objects are assumed to be distributed randomly with a spatial Poisson distribution characterized by a given constant density. Four different types of features have been identified and modeled: grass plots, brush plots, tree trunks, and crowns of trees. Although all features have the same geometry, they are distinguished by types within a vegetation class, according to their attributes. For instance, grass plots tend to have small heights and large widths, whereas three trunks usually have large heights, but small widths.

Vegetation modeling is accomplished for two main reasons. The first is concerned with the obscurative effects that various vegetation features exert on line of sight determination. The second involves degradative effects caused by certain vegetation features which retard the movement of certain equipment types; this in turn influences the rate of movement assigned to a unit.

Given an observer-target pair within the area of operation, the line of sight verdict between them depends mainly on the terrain features of relief and vegetation. To conserve processing time, relief is always checked first. Obviously if it is determined that relief interrupts the line of sight, the effects of vegetation need not be considered. Otherwise, vegetation effects are examined in a probabilistic fashion. The end result is a line of sight verdict represented in terms of an integer percent indicating the expected fraction of the target unit visible to the observer unit. This expected fraction is computed based upon several factors:

1) The heights, widths, and densities of feature types in the intervening classes of vegetation between the observer and the target

2) The distance between the observer and target

3) The typical height and width of the single element used, for detection purposes, to represent the target unit.

Movement of certain types of equipment is affected by the class of vegetation in which the equipment is situated. The prominent vegetation feature retarding equipment mobility was determined to be tree trunks. Two attributes of tree trunks are responsible for the degradation:

1) The width (i.e., diameter) of the tree trunks

2) The density of the tree trunks (which determines the mean spacing between them).

Width of tree trunks is a significant factor, because certain equipment types can overturn tree trunks (i.e., totally ignoring them), whereas others must maneuver around them. The amount of maneuvering depends upon the density of tree trunks in the area. Clearly, movement is hampered more severely when operating in a dense environment.

4.22.2 Programmatical Description

The Vegetation Class Data Table is composed of the FORTRAN arrays H(I,J), W(I,J), and RHO(I,J). Data describing a vegetation class is referenced by specifying the class number I, where I ranges in value from 1 to 16 inclusively. Each class is comprised of four different feature types; they can be referenced by the index J ($1 \leq J \leq 4$). Thus to obtain data describing an attribute of vegetation, the class number and feature type index must be known.

Arrays H and W contain the heights and widths respectively of the (right) circular cylindrical solids used to model each feature type. Height and width data are in units of meters. The array RHO specifies the densities of each feature type and is expressed in terms of numbers of objects in a square having sides fifty meters in length. The structure and organization of the Vegetation Class Data Table is illustrated in Table 4-11 .

Table 4-11 . Vegetation Class Data Table

VEGETATION CLASS	FEATURE TYPE DESCRIPTION J	ATTRIBUTE 1 HEIGHT (METERS)	ATTRIBUTE 2 WIDTH (METERS)	ATTRIBUTE 3 DENSITY (OBJECTS/50 METER SQ)
1	GRASS PLOTS J=1 BRUSH PLOTS J=2 TREE TRUNKS J=3 TREE CROWNS J=4	H(1,1) H(1,2) H(1,3) H(1,4)	W(1,1) W(1,2) W(1,3) W(1,4)	RHO(1,1) RHO(1,2) RHO(1,3) RHO(1,4)
2	GRASS PLOTS J=1 BRUSH PLOTS J=2 TREE TRUNKS J=3 TREE CROWNS J=4	H(2,1) H(2,2) H(2,3) H(2,4)	W(2,1) W(2,2) W(2,3) W(2,4)	RHO(2,1) RHO(2,2) RHO(2,3) RHO(2,4)
. . . .	. . . .	. . . .	. . . .	. . . .
N	GRASS PLOTS J=1 BRUSH PLOTS J=2 TREE TRUNKS J=3 TREE CROWNS J=4	H(N,1) H(N,2) H(N,3) H(N,4)	W(N,1) W(N,2) W(N,3) W(N,4)	RHO(N,1) RHO(N,2) RHO(N,3) RHO(N,4)
. . . .	. . . .	. . . .	. . . .	. . . .
16	GRASS PLOTS J=1 BRUSH PLOTS J=2 TREE TRUNKS J=3 TREE CROWNS J=4	H(16,1) H(16,2) H(16,3) H(16,4)	W(16,1) W(16,2) W(16,3) W(16,4)	RHO(16,1) RHO(16,2) RHO(16,3) RHO(16,4)

Note the size limitation of the table. Currently a maximum of sixteen different classes may be defined, and within each class no more than four feature types may be distinguished. The data entered into the above arrays is done during model initialization; this data remains constant throughout the simulation. Subroutine LOSINP inputs the data using the XEROX BUFFER IN operation. As mentioned before, the arrays are static, and their contents are not modified by any subroutine; the data however is referenced by the following subroutines (for line of sight processing):

LOSINP

LOSVEG

VEGCON.

Caution must be extended when modelling movement degradation. Unfortunately the overlay structure does not allow the movement degradation submodule to access data from the terrain/line of sight module. This means that vegetation data must be input such that it is available for use in the overlay containing the movement degradation code (subroutine ADJROM). Since the only feature type that affects movement is tree trunks, only data defining the attributes of this type need to be copied. This is accomplished by defining two new arrays, DIAMTER and DENSITY, to store width and density data (height data is not needed). Each array contains sixteen words of storage; the arrays are initialized by the NAME LIST convention. Width data stored into array DIAMTER should be exactly the same as that contained in array W. Density however must be stored in units of objects per square meter rather than objects per fifty meter square (the necessary conversion is obtained by dividing the appropriate elements of the RHO array by 2500.0). Arrays DIAMTER(16) and DENSITY(16) comprise common block TREEDTA and the data contained in it is related as follows:

Vegetation Class	Width of Tree Trunks DIAMTER	Density of Tree Trunks DENSITY
1	DIAMTER(1) = W(1,3)	DENSITY(1) = RHO(1,3)/2500
2	DIAMTER(2) = W(2,3)	DENSITY(2) = RHO(2,3)/2500
.	.	.
.	.	.
.	.	.
N	DIAMTER(N) = W(N,3)	DENSITY(N) = RHO(N,3)/2500
.	.	.
.	.	.
.	.	.
16	DIAMTER(16)= W(16,3)	DENSITY(16)= RHO(16,3)/2500

The data in these arrays remains constant throughout the simulation; data is referenced but not modified by the following subroutine:

ADJROM.

4.23 SUPPRESSION CURVE TABLE

The Suppression Curve Table consists of the following array:

SIGTAB (10,6)

4.23.1 English-language Description

The Suppression Curve Table is closely related to the Suppression Criterion Selection Table. The outcome of a unit searching the Suppression Criterion Selection Table is the selection of 1) a criterion and 2) a curve which together are used to compute the fraction that the unit is suppressed. The criterion is evaluated, and that value is used against the designated suppression curve. Through linear interpolation, the level of suppression is determined as a direct function of the value of the criterion. Two examples are given in the Programmatical Description. Up to ten curves can be defined in CATTS at initialization time, with four points defined for each, as shown in Figure 4-3 .

4.23.2 Programmatical Description

The calculation of a unit's suppression by means of the Suppression Curve Table is discussed by means of two examples which use actual CATTS FEBA Gold Scenario Data.

Figures 4-4 and 4-5 represent the four different suppression curves in FEBA Gold. Six data base values identify the four points defining each curve, in the following format:

Value #1 - first ordinate
2 - second abscissa
3 - second ordinate
4 - third abscissa
5 - third ordinate
6 - fourth abscissa

the first abscissa is assumed to be zero
the fourth ordinate is assumed to be 1.0

Two examples follow to illustrate the accessing of a specific entry in the table, and the subsequent computations required to compute the suppression factor.

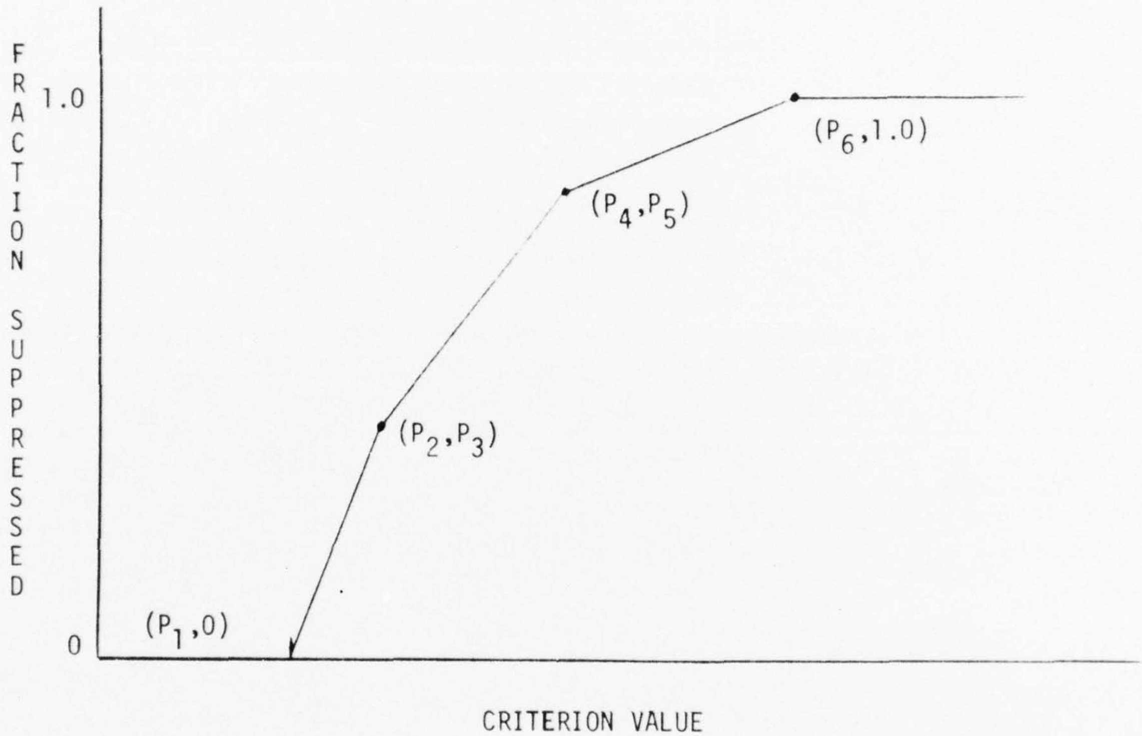


Figure 4-3 . Six Point Function Table for Determining Suppression

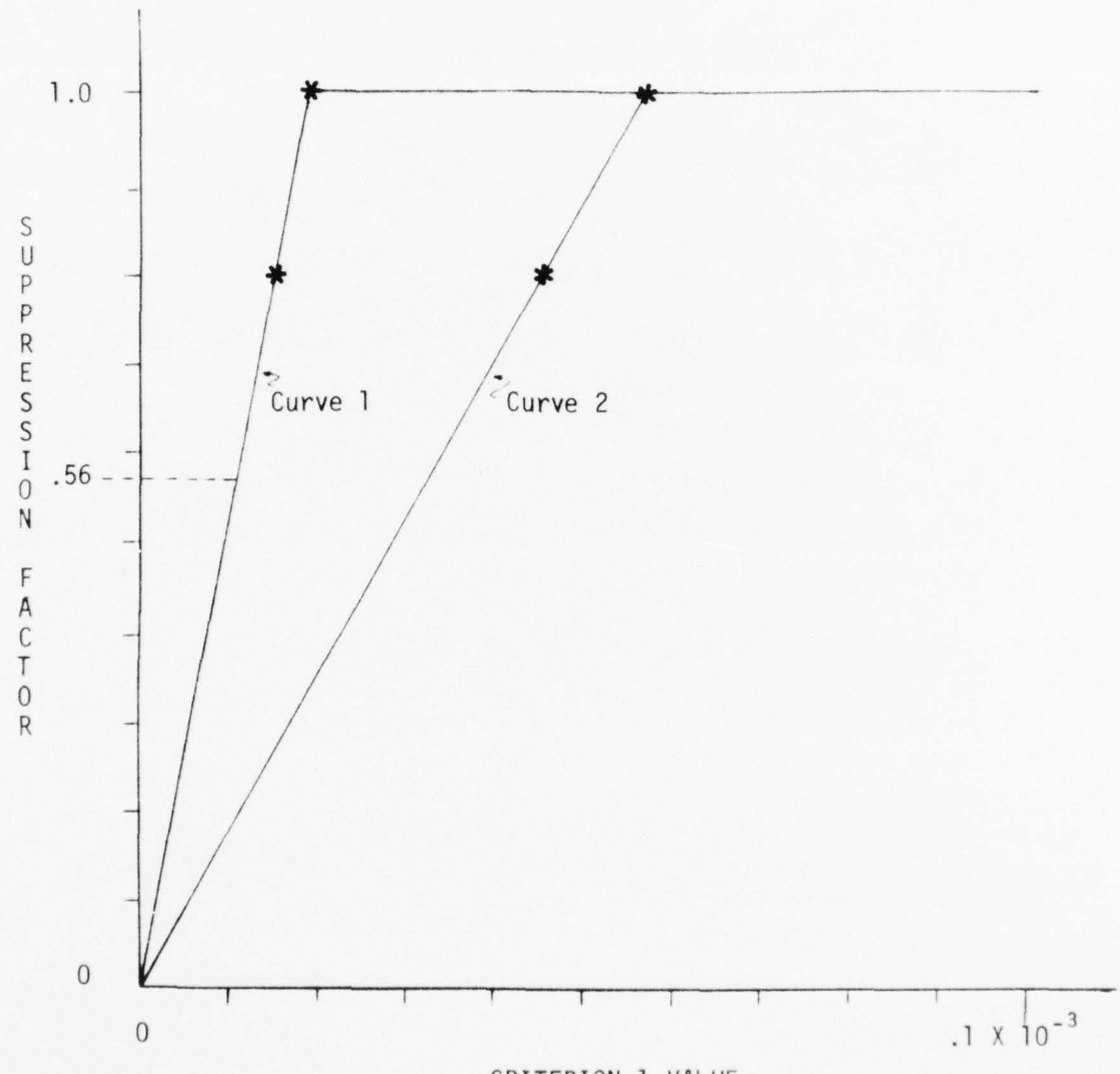


Figure 4-4 Suppression Curves in FEBA Gold

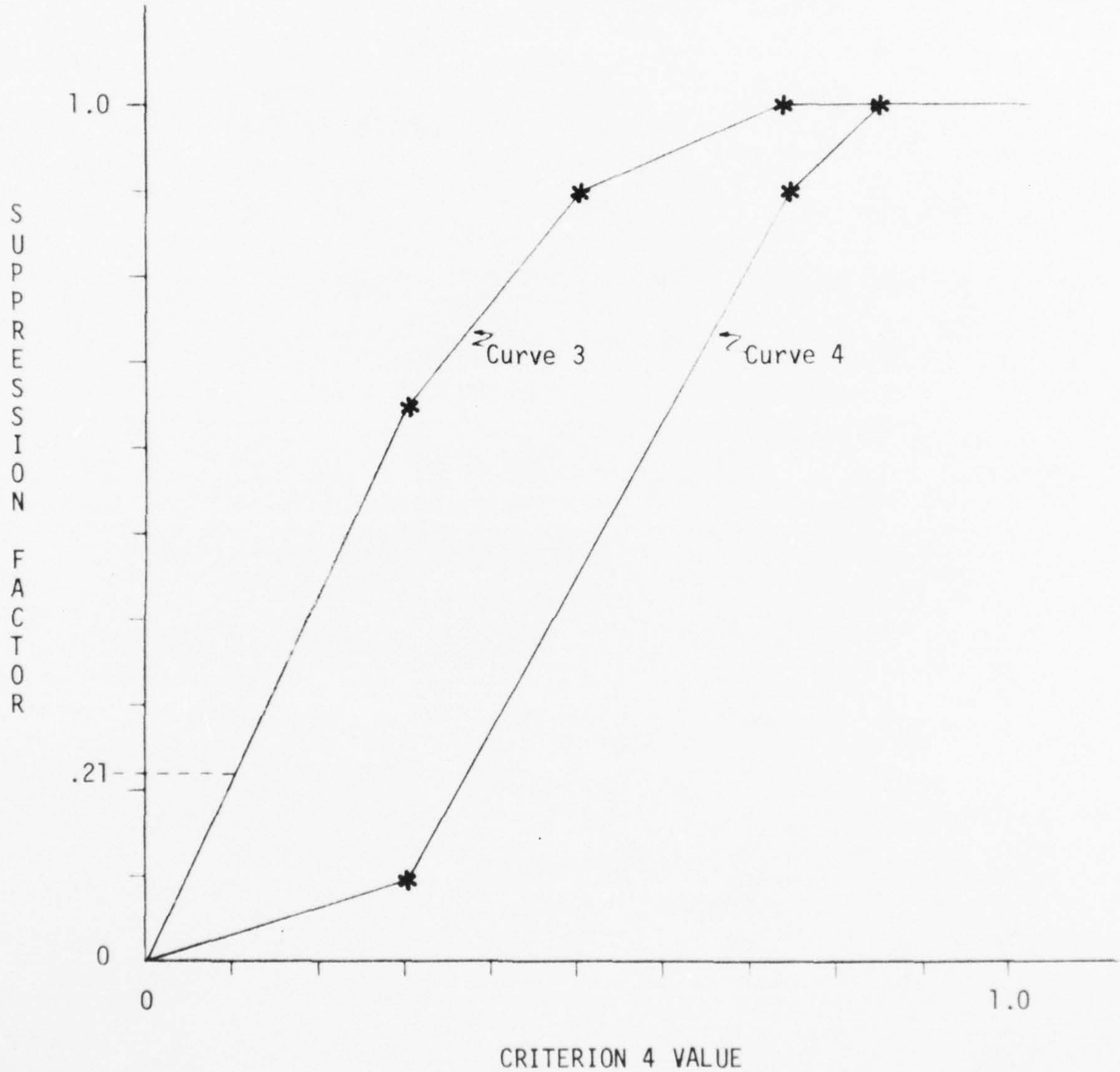


Figure 4-5 Suppression Curves in FEBA Gold

The suppression data table in the current FEBA Gold Scenario is condensed into Table 4-12 below for use in the examples. The actual 8 digit entry in the data base table is in the following format:

C = number of suppression criterion
XX = unit type (0 = > any type)
YY = unit operational state (0 = > any state)
Z = Red (1) or Blue (2) (0 = > either)
SS = number of suppression curve to be used.

Example 1.

A Blue "mechanized infantry" unit (type 1) in the state of "moving to contact" (op. state 11) determines his suppression factor using entry 1 of the summarized suppression table by calculating the equation given in criterion 4, and applying the result to curve #3. If this unit began the exercise with 100 personnel, and has accumulated 10 casualties to date, the criterion 4 equation yields a value of .1, which is then used as the abscissa of curve #3. The corresponding ordinate of .21 is the fraction of the unit suppressed.

Example 2.

A Red artillery unit (type 7) in the state of firing at "close support - normal" (op. state 64) determines his suppression factory using entry 5 of the summarized suppression table by calculating the equation given in criterion 1, and applying the result to curve #1. If this unit received 4 rounds of artillery in the last time step (SWFU(I)=4) and the unit is 200 meters square (unit area = 40000 m^2), the criterion 1 equation yields a value of .0001, which is then used as the abscissa of curve #1. The corresponding ordinate of .56 is the fraction of the unit suppressed.

Table 4-12 . Suppression Table (Summarized)

Entry	Criterion #	Unit Type	Unit Op. State	Red or Blue	Curve #
→ 1	4	1-4	11-17	Both	3
2	4	1-4	21,22	Both	4
3	1	5	41-44	RED	1
4	1	5	41-44	BLUE	2
→ 5	1	any	51-59, 61-69	RED	1
6	1	any	51-59, 61-69	BLUE	2
7	1	8	71-74	RED	1
8	1	8	71-74	BLUE	2
9	4	any	31-34	Both	4
10	1	any	any	RED	1
11	1	any	any	BLUE	[illegible]

AD-A038 796 TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF F/G 15/7
MATHEMATICAL MODEL USER'S MANUAL COMBINED ARMS TACTICAL TRAININ--ETC(U)
JAN 77 D S ADAMSON, E C ANDREANI, G W ARCHER N61339-73-C-0156

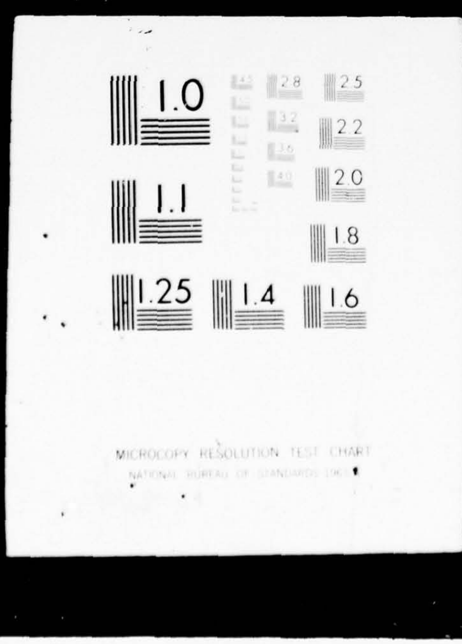
UNCLASSIFIED NAVTRAEQUIPC-73-C-0156-E00 NL

8 OF 8
AD A038796

END
DATE FILMED
5-77

8 OF 8

AD
A038796



4.24 SOIL CLASS DATA TABLE

The Soil Class Data Table contains data used to represent the terrain feature of soil; the data is stored in the FORTRAN array:

SOIL (15, 13).

4.24.1 English-language Description

The Soil Class Data Table contains numerical values which are used to model the different types of soil in the area of operation. Currently the size of the table will allow a maximum of 15 different classes of soil to be modeled concurrently. As far as the CATTS model is concerned, soil affects visual detection (because of the reflectance property of the soil type), detection verdicts from unattended ground sensing devices, and equipment movement. The effect on equipment movement ultimately influences the movement rate of the unit. In short, soil is an entity defined by the following attributes:

1) Rating cone index (wet and dry) affecting equipment movement
2) Detection degradation factors affecting unattended ground sensing devices
3) Luminous reflectance factor affecting visual detection.

The Soil Data Table stores data which models the attributes described above.

Every type of soil has a number associated with it which characterizes its trafficability by vehicles. This number is called the Rating Cone Index (RCI) of the soil. It is determined mainly by the shear strength of the soil (i.e., the resistance of soil particles to movement across other soil particles) under pressure, like that exerted by vehicular traffic. The RCI is usually given by a dimensionless number, ranging in value from 0.0 to about 170.0. The higher the value, the better the soil for vehicular travel. Soil strength is affected by the moisture content, hence the table presents two sets of Rating Cone Indices, one for dry soil, and the other for soil saturated with moisture.

Ten different types of unattended ground sensing devices are modeled. The performances of such devices is affected by the type of soil that it is located in. In general, soil tends to degrade the operations of ground sensing devices. The degree of degradation is given by a fraction, ranging in value

from 0.0 to 1.0 inclusively. A fraction tending toward 0.0 indicates that the device is adversely affected by the soil it is located in. Thus for each type of soil represented, ten fractions are included to reflect its effect on the operation of the ten unattended sensor devices being modeled.

The luminous reflectance for various background features and materials is substantially different and is a very important factor to be considered in determining daytime visual detection verdicts. Every soil class being modeled has a fractional value representing its reflectance. This value is used, along with other data describing surrounding terrain features, to compute the apparent contrast between a target unit and its background. Apparent contrast in turn is used to determine contrast ratio which gives rise to a single-glimpse probability of detection. Other factors are considered (i.e., optical devices, human alertness, etc.) before a final visual detection probability is established. For details, the reader should refer to the model description of Target Acquisition Module (Section 5.3).

4.24.2 Programmatical Description

The FORTRAN array SOIL(I,J) contains data relating the effects of various soil classes on equipment movement and detection. Currently, a maximum of 15 soil classes may be defined. The array is referenced by soil class number I, where I may range from 1 to 15 inclusively. Thirteen words of data, referenced by the index J ($1 \leq J \leq 13$), are associated with each soil class. The data is entered using the XEROX NAMELIST convention.

As Table **4-13** illustrates, the array is comprised of fifteen rows of data, one row for each soil class modeled, and thirteen columns, each column containing a numerical value specifying a soil characteristic. Column 1 of the array contains the Rating Cone Indices of the various classes of soil when dry. Each Rating Cone Index is a positive number ranging from 0.0 to approximately 170.0. A high index reflects a high degree of trafficability by vehicles. Column 2 of the array contains the Rating Cone Indices when the soil type is wet. Columns 3 through 12 contain performance degradation factors for unattended ground sensors. These factors are fractions having values between zero and one inclusively, where values tending toward zero indicate higher degrees of degradation. Column 13 contains fractions, also inclusively between zero and one, which model the factor of luminous

Table 4-13 . Soil Class Data Table as Given by Array SOIL

SOIL CLASS I \ WORD J	1	2	3	4	5	6	7	8	9	10	11	12	13
1	R	R											L
2	A T	A T											U M
3	I N	I N											I N
4	G	G											O U
5	C O	C O											S
6	N E	N E		PERFORMANCE DEGRADATION FACTORS									R
7	I	I		FOR UNATTENDED GROUND SENSORS (U.G.S.) OF TYPES 1 THROUGH 10									E F
8	N D	N D											L E
9	I C	I C											C T
10	E S	E S											A N
11	W	W											C E
12	H E	H E											V
13	N	N											A L
14	D R	W E											U E
15	Y	T											S

reflectance. This factor is required by the visual detection logic. The following summarizes the data required to model soil:

SOIL(I,J) = Characteristic data for soil class I ($1 \leq I \leq 15$):

J=1 for RCI when soil is dry

J=2 for RCI when soil is wet

J=3 performance degradation factor for UGS type 1

J=4 performance degradation factor for UGS type 2

. . . .

. . . .

. . . .

. . . .

. . . .

J=12 performance degradation factor for UGS type 10

J=13 luminous reflectance of soil class I.

SOIL is a static array; this means that once user defined data has been entered into the array, no other routines modify the data. The Soil Class Data Table remains constant throughout the simulation. Data is entered via three different sources. Routine FORMAIN utilizes a DATA statement to enter default values into the array. Then subroutine INPUT calls subroutine SENSINP to read in data uniquely specified for a particular scenario; this data overrides the default values. Finally, the array values can be overriden once more. This is done by reading in data values from the NAME LIST; this option allows the user to easily vary soil class data when running the same scenario several times.

The data found in array SOIL is utilized by the movement and detection logic of the CATTS math model. Movement is concerned with trafficability of vehicular equipment over various classes of soil. Subroutine ADJROM accesses the RCI data from the array to derive movement degradation factors. The detection logic deals with the obscurative effect of soil on daytime visual capabilities, as well as the reduced performance of unattended sensing devices due to soil characteristics; these are handled by subroutines VISUAL and UASCK respectively. The following is a list of all routines making reference to the array SOIL:

ADJROM	UASCK
FORMAIN	VISUAL.
SENSINP	

4.25 CASUALTY STATISTICS TABLE

The Casualty Statistics Table consists of the following array:

STATS (40,41,2)

4.25.1 English-language Description

The Casualty Statistics Table is used to accumulate personnel and equipment casualties from the beginning of the exercise to its completion. The accumulation is performed for each weapon of the opposition, for both Red and Blue.

At the beginning of the exercise, the list of ground fire equipment is determined for each side, up to a maximum of 38 weapons. This list does not change throughout the exercise. A row of the table is reserved for each equipment to store personnel and equipment casualties attributed to it. As opposing personnel and equipment casualties are produced during the exercise as a result of firing a particular equipment (weapon), the appropriate row of entries for the weapon are accumulated. Two additional rows are set aside in the table for other types of casualties. The first is used to store accumulated casualties resulting from all other casualty-producing causes, including:

- self-destroyed equipment that could not be manned
- equipment lost due to maintenance attrition
- personnel and equipment casualties due to a minefield encounter.

The table is cleared at model initialization. Every casualty produced in a given minute of simulation time is accumulated in its appropriate row in either the Red or Blue side of the table, as a result of either firing a ground weapon (Ground Fire Module), delivering air ordnance (Air Module), or incurring one of the three other possibilities listed above. All casualty values are integers when they are accumulated in the table.

At the completion of the exercise, a casualty report is printed by the line printer from the data in the Casualty Statistics Table. The format of the report is depicted in Table 4-14 . An equipment (weapon)

Table 4-14. Casualty Report Format

"Blue Personnel and Equipment Losses by Weapon Type at Time"

		1	2	. . .	Last	39	40	41
	Blue → / Red ↓	First Blue Eqpt	Second Blue Eqpt	. . .	Last Blue Eqpt	Not Used	Not Used	Blue Personnel
1	First Red Weapon							
2	Second Red Weapon							
.	.	.	.	. . .	.			.
.	.	.	.	. . .	.			.
.	.	.	.	. . .	.			.
Last	Last Red Weapon							
Last + 1	"Air Losses"							
Last + 2	"Other Losses"							

TOTALS			. . .				

"Red Personnel and Equipment Losses by Weapon Type at Time"

		1	2	. . .	Last	39	40	41
	Red → / Blue ↓	First Red Eqpt	Second Red Eqpt	. . .	Last Red Eqpt	Not Used	Not Used	Red Personnel
1	First Blue Weapon							
2	Second Blue Weapon							
.	.	.	.	. . .	.			.
.	.	.	.	. . .	.			.
.	.	.	.	. . .	.			.
Last	Last Blue Weapon							
Last + 1	"Air Losses"							
Last + 2	"Other Losses"							

TOTALS			. . .				

causing no casualties is omitted from the report.

4.25.2 Programmatical Description

The Casualty Statistics Table, array STATS, is initialized by subroutine INPUT. The initialization procedure consists of clearing all table entries (actually done in subroutine INIT) and setting up the table row and column indices. The STATS array is three dimensional, STATS (I,J,K), of size STATS (40,41,2). For K = 1, the table values contain casualty statistics for Red firing against Blue; for K = 2, Blue firing against Red. When k = 1 (Red firing), index I ($1 \leq I \leq 38$) points to the Ith Red equipment, stored in array NDXSTATS (I,K), ($1 \leq I \leq 38$, $1 \leq K \leq 2$), and index J points to the Jth Blue equipment in NDXSTATS (k = 2 indicates the list of Blue equipment). Two values of I following the last weapon are reserved for "air losses" and "other losses", respectively, as mentioned in 4.25.1. Two values of J, J = 39 and 40, are not used, and J = 41 is reserved for personnel casualties.

At initialization, the row and column indices, I and J, are determined by listing all equipments used in any Red unit in array NDXSTATS (IX,1), and all equipments used in any Blue unit in array NDXSTATS (IX,2), up to a maximum of 38 equipments for each side. If more than 38 are used in the model, a RAMALERT is generated and displayed on the TV monitor. The array NDXSTATS is actually dimensioned 40 * 2, but the last two slots are not used for both Red and Blue. For a given color, say Red, the list of equipments in NDXSTATS (IX,1) is made by searching the equipment list of each Red unit, and whenever a new equipment is found, an entry is made in the array. The Blue equipment list, NDXSTATS (IX,2), is composed similarly. The lists, therefore, bear no relation to equipment number. Consequently, a cross-referencing scheme is used by means of array NDXEQ (IEQ,K), dimensioned 80 * 2, where IEW is the equipment number and K = 1 denotes Red, K = 2 denotes Blue. NDXEQ (IEQ,1) contains the index, IX, into the NDXSTATS (IX,1) array, which is a list of Red equipments; likewise for NDXEQ (IEQ,2). These arrays, once determined, are not altered during an exercise. An example taken from the composition of several units in the

FEBA Gold scenario follows to illustrate the determination and use of the indices to the STATS array.

Red Units	Equipment List
1 (A/1/107)	2,3,4,5,6,7,10,11,18,44,1
2 (B/1/107)	2,3,4,5,6,7,10,11,18,1
3 (C/1/107)	2,3,4,5,8,9,11,18,20,1

Blue Units	Equipment List
45 (HV MOR/2-77)	34,32,40,47,48,50,26
46 (AT/2-77)	30,32,40,50,26
49 (1/A/2077)	27,29,32,47,50,26

IX	NDXSTATS (IX,1)	
1	2	NDXEQ(2,1) = 1
2	3	NDXEQ(3,1) = 2
3	4	NDXEQ(4,1) = 3
4	5	NDXEQ(5,1) = 4
5	6	NDXEQ(6,1) = 5
6	7	NDXEQ(7,1) = 6
7	10	NDXEQ(10,1) = 7
8	11	NDXEQ(11,1) = 8
9	18	NDXEQ(18,1) = 9
10	44	NDXEQ(44,1) =10
11	1	NDXEQ(1,1) =11
12	8	NDXEQ(8,1) =12
13	9	NDXEQ(9,1) =13
14	20	NDXEQ(20,1) =14

IX	NDXSTATS (IX,2)	
1	34	NDXEQ(34,2) = 1
2	32	NDXEQ(32,2) = 2
3	40	NDXEQ(40,2) = 3
4	47	NDXEQ(47,2) = 4
5	48	NDXEQ(48,2) = 5
6	50	NDXEQ(50,2) = 6
7	26	NDXEQ(26,2) = 7
8	27	NDXEQ(27,2) = 8
9	29	NDXEQ(29,2) = 9

After initialization, every casualty produced during a simulated minute of activity is added into the appropriate element of the STATS array. For example, if Red equipment type 10 (tank) destroys a Blue equipment type 40 (truck), the casualty of one is added to element STATS (NDXEQ(10,1), NDXEQ(40,2), 1), which, in the above example is STATS (7,3,1).

The following subroutines add casualties to the STATS array for the reasons shown:

ADW	Air ordnance delivery
OBSDELAY	Minefield encounter
REDIST	Self-destruction of unmanned equipment
STEP	Maintenance attrition
WPNFIR	Ground fire

At the conclusion of an exercise subroutine CASREP prints a casualty report in the format shown in Table 4-14 . In addition to listing and identifying the data in the STATS table, the subroutine accumulates the casualty total for each column over all inflicting weapons and other casualty-producing causes. These results constitute a measure of success for the Red and Blue forces.

4.26 MODE DISTRIBUTION VECTOR TABLE

4.26.1 English-language Description

Each equipment type may be operated in up to eight different modes. Each mode is defined in terms of four parameters; rate of movement, ammunition type, rate of fire (for weapons), and vulnerability of personnel operating the equipment to enemy fire.

The parameters are defined in such a way that, for non-firing, direct, and indirect firing equipments, the eight categories represent the following modes of operation for the equipment:

Mode	Characteristics: Move Rate	Fire Rate	Personnel * Vulnerability
1 (suppressed)	0	0	1
2	Maximum	0	1 - 4
3	Fast	0	1 - 4
4	Medium	0	1 - 4
5	Slow	0	1 - 4
6	0	Sustained	1 - 4
7	0	Maximum	1 - 4
8	0	0	1 - 4

For support fire weapons, the modes represent a geographic classification of opposing target units, where the fire rates are defined specifically for each target class. The modes are defined as follows:

---

* Personnel Vulnerability Classes are defined as:

1 = unsuppressed

2 = foxhole

3 = prone

4 = standing

Mode	Move Rate	Fire Rate	Personnel * Vulnerability
		Characteristics	
1 (suppressed)	None	0	1
2	Medium	0	1 - 4
3 (close support)	0	C.S. Normal	1 - 4
4 (close support)	0	Final Protective Fire	1 - 4
5 (interdictory)	0	Interdictory	1 - 4
6 (op. groups)	0	Op. Groups	1 - 4
7 (counter-battery)	0	Counter-battery	1 - 4
8 (general)	0	General	1 - 4

The selection of weapon effects function also reflects the mode of weapon operation (see Weapon Effects Look-up Table in this section).

Equipment of a given type in a unit may be distributed over several modes of operation at the same time. The distribution of this equipment among its various modes is specified by the mode distribution vector, which is a matrix containing eight fractions between zero and one. Each of these numbers is associated with an operating mode of the equipment and prescribes

---

* Personal Vulnerability Classes are defined as:

1 = unsuppressed
2 = foxhole
3 = prone
4 = standing

what fraction of the equipment is to operate in that mode when the unit is unsuppressed. The fraction of men and equipment in each mode is further reduced proportionally to the degree of suppression in the unit. If an equipment type operates in fewer than eight different modes, the unused spaces in the mode distribution vector are set to zero.

A set of up to 80 different mode distribution vectors may be specified by input and stored for use in the Mode Distribution Vector Table for use during the exercise of the model. The particular vector of this set that is used for direct, indirect, and non-firing equipment types of a unit at any given time is determined by three factors: the operational state of the unit, the nature of different enemy units that the equipment will be employed against if it is a weapon, and the proximity of enemy units. The first two factors are embodied in the mode selection code (see Mode Selection Code Table in this section). The third factor involves the mode selection range, MUSLRA, which is compared against the break ranges in the Break Range Table, discussed elsewhere in this section.

For support fire weapons, the Fire Support Table directly specifies which mode distribution vector is to be used for a given support fire weapon in a unit. See the discussion for the Fire Support Table.

As the various modes of operation of different equipment types are determined, information on the vulnerability of operating personnel in the fire unit is accumulated, thus providing a vulnerability profile that can be used in the <u>next</u> time step in computing the effects of enemy fire on the fire unit.

See the definition for the term "mode" in Section 3.0 for related information.

4.26.2 Programmatical Description

The Mode Distribution Vector Table, represented in the model by array XMUTAB (80,8), is loaded with input values from the data base at model initialization, and thereafter is referenced every time step. The eight entries of a selected row, I, represented by XMUTAB (I,1→8), comprise the

elements of mode distribution vector I. Every non-firing equipment, direct or indirect fire weapon in every unit is processed by the Ground Fire Module. As part of that processing, subroutine ORGFIR makes successive calls, first to SCHRMU to determine the appropriate Break Range Table row as a result of a Model Selection Code Table match, then to SCHMU to evaluate the Break Range Table row to yield a specific mode distribution vector number. This number is used by ORGFIR itself as the first (row) index into array XMUTAB. The number of pieces of the equipment manned in the unit are then distributed over the eight modes of equipment operations in proportion to the eight fractional values in the selected row of XMUTAB. The following example illustrates this procedure.

Example

Suppose a Red unit 33% suppressed is having its 10 tanks processed by the Ground Fire Module, and the processing of the Mode Selection Code Table and the Break Range Table results in mode distribution vector 13 being selected. From the current FEBA Gold scenario, XMUTAB (13,1-8) is

	(13,1)	(13,2)	(13,3)	(13,4)	(13,5)	(13,6)	(13,7)	(13,8)
XMUTAB	0	0	0	0	.67	0	.33	0

The computations of subroutine ORGFIR result in the following successive allocations of the tanks across the eight modes:

Before Suppression Factor

Mode	1	2	3	4	5	6	7	8
# of Tanks	0	0	0	0	6.7	0	3.3	0

After Suppression Factor

Mode	1	2	3	4	5	6	7	8
# of Tanks	3.3	0	0	0	4.5	0	2.2	0

Since mode 1 is totally suppressed, mode 5 is the slow movement rate (no firing), and mode 7 is the maximum firing rate, the following effects will result:

- 2.2 of the 10 tanks will fire at the maximum firing rate.
- 4.5 of the 10 tanks will contribute to the equipment's, and ultimately, the unit's movement rate.
- all personnel manning the tanks (usually 3 per tank for a total of 30) are placed in the invulnerable class (#1) by means of the data associated with tanks in FEBA Gold.
- ammunition type 10 is fired from the tank.

Every non-firing equipment, direct and indirect fire weapon in every unit for both Red and Blue forces is distributed similarly.

Support fire weapons are also processed similarly after the mode distribution vecotr number has been selected. However, the Fire Support Table yields the mode distribution vector directly.

The three current scenarios, FEBA Gold, FEBA Silver and Attack use 43 of the 80 available rows in XMUTAB. The sum of each row must equal 1.0.

4.27 VEGETATION POLYGON TABLE

The Vegetation Polygon Table consists of the following arrays: ICL(225), ITRC(225), XPOLY(5, 225), YPOLY(5, 255).

4.27.1 English-language Description

The Vegetation Polygon Table is used to partition the area of operation into disjoint regions having classes of vegetation which differ from that of the dominant class. Currently sixteen different classes of vegetation are simulated. The class of vegetation which is most typical of that found in the area of operation should be chosen as the dominant class. All areas which do not contain the dominant vegetation should be traced out, in approximate fashion, to fit into the following forms:

1) Triangle
2) Rectangle
3) Circle.

Each polygonal form has associated with it data identifying the geometrical form, the non-dominant class of vegetation it is tracing out, and the X-Y coordinate values specifying the polygon location and dimensions. Currently, a maximum of 225 disjoint polygons may be specified to more accurately simulate global vegetation.

The data in the Vegetation Polygon Table is used by the Line of Sight/Terrain Submodule for two main purposes. The first involves the simulation of obscurative effects produced by intervening vegetation when an observer unit attempts to visually detect a given enemy unit. The line of sight between the two units will usually pass through several different classes of vegetation. Each class contains different feature types of varying size and density; this produces various degrees of concealment and affects the eventual line of sight determination.

The second purpose deals with determining the class of vegetation each unit is situated in. This data is required by the movement degradation submodule to model the retarding effects of certain vegetation features on equipment types contained in a unit. It has been determined that width and density of tree trunks are prominent factors which affect equipment movement. Of course any factor which affects equipment movement, eventually affects overall unit movement rate. The

detection module also requires knowledge of the surrounding vegetation characteristics. In particular, factors like background noise level for aural detection and luminous reflectance for visual detection differ between various vegetation classes.

4.27.2 Programmatical Description

The FORTRAN arrays ICL(I), ITRC(I), XPOLY(J,I), and XPOLY(J,I) contain data making up the Vegetation Polygon Table. Each array is indexed by polygon number I, where I is an integer ranging in value from 1 to 225 inclusively. Note the size limitation on the arrays; a maximum of 225 polygons may be described to partition the area of operation into non-dominant vegetation regions. The arrays XPOLY(J,I) and YPOLY(J,I) are also indexed by J, where J is an integer ranging in value from 1 to 5 inclusively. It references storage locations for XY coordinate values describing the I-th polygon. The structure of the Vegetation Polygon Table is illustrated in Table 4-15 .

The above arrays are initialized at the start of the simulation. Subroutine LOSINP utilizes the XEROX BUFFER IN operation to read in user defined data which trace out all regions of non-dominant vegetation. This initial data remains constant throughout the simulation and should not be modified by any subroutine. All information about a particular polygon can be retrieved by specifying the polygon number I. The I-th element of the array ICL should contain a code indicating which class of vegetation is being represented by the I-th polygon. The code is an integer which can take on a value between 1 and 16 inclusively (there are 16 different classes of vegetation modeled). The array ITRC contains a code in its I-th element designating the geometric form used to trace out the region. This code is an integer with the following range of values:

1 = triangle

2 = rectangle

3 = circle.

The I-th elements in the arrays XPOLY and YPOLY contain XY coordinate values and dimension data detailing the I-th polygon.

The data organization in the arrays XPOLY and XPOLY follows specific conventions which differ according to whether the polygon is a triangle, rectangle, or circle. If the I-th polygon is a triangle, the XY coordinates of its three vertices are given in the array elements of XPOLY and YPOLY. Specifically, using

Table 4-15 . Vegetation Polygon Table

POLYGON NUMBER	VEGETATION CLASS	GEOMETRIC FORM	XY COORDINATES	
1	ICL(1)	ITRC(1)	XPOLY(1,1) XPOLY(2,1) XPOLY(3,1) XPOLY(4,1) XPOLY(5,1)	YPOLY(1,1) YPOLY(2,1) YPOLY(3,1) YPOLY(4,1) YPOLY(5,1)
2	ICL(2)	ITRC(2)	XPOLY(1,2) XPOLY(2,2) XPOLY(3,12) XPOLY(4,2) XPOLY(5,2)	YPOLY(1,2) YPOLY(2,2) YPOLY(3,2) YPOLY(4,2) YPOLY(5,2)
⋮	⋮	⋮	⋮	⋮
I	ICL(I)	ITRC(I)	XPOLY(1,I) XPOLY(2,I) XPOLY(3,I) XPOLY(4,I) XPOLY(5,I)	YPOLY(1,I) YPOLY(2,I) YPOLY(3,I) YPOLY(4,I) YPOLY(5,I)
⋮	⋮	⋮	⋮	⋮
225	ICL(225)		XPOLY(1,225) XPOLY(2,225) XPOLY(3,225) XPOLY(4,225) XPOLY(5,225)	YPOLY(1,225) YPOLY(2,225) YPOLY(3,225) YPOLY(4,225) YPOLY(5,225)

the standard Cartesian coordinate system, data describing the triangle should be entered in this order:

$\{XPOLY(1,I),\ YPOLY(1,I)\}$ = (X,Y) of the lowest vertex

$\{XPOLY(2,I),\ YPOLY(2,I)\}$ = (X,Y) of the middle vertex

$\{XPOLY(3,I),\ YPOLY(3,I)\}$ = (X,Y) of the highest vertex.

Ambiguities arise however, if the I-th polygon is a triangle having a horizontal leg (i.e., one side of the triangle is parallel with the X-axis). When this condition exists, the triangle will contain either (1) two "lowest" vertices, or (2) two "highest" vertices. The ambiguities are resolved by adopting the convention that the left-most (i.e., closest to Y-axis) vertex is to be designated the "lowest" vertex in the former case and the "highest" vertex in the latter case.

If the I-th polygon is a rectangle, the arrays XPOLY and YPOLY will contain the XY coordinates of two of its vertices and a given length. The software will generate the X-Y coordinates of the other two vertices whenever the need arises. Thus the following convention is adopted:

$\{XPOLY(1,I),\ YPOLY(1,I)\}$ = (X,Y) of the lower vertex

$\{XPOLY(2,I),\ YPOLY(2,I)\}$ = (X,Y) of the left-most vertex

$\{XPOLY(3,I)\}$ = length of the rectangle; used to generate the other two vertices when required.

If the I-th polygon is a circle, the array XPOLY and YPOLY will contain the XY coordinates of the center of the circle and its radius. Circles have the simpliest organization:

$\{XPOLY(1,I),\ YPOLY(1,I)\}$ = (X,Y) of center of circle

$\{XPOLY(3,I)\}$ = radius of circle.

The data organization according to geometric form for the I-th element in arrays XPOLY and YPOLY are presented below:

GEOMETRIC FORM / ARRAY ELEMENT	TRIANGLE	RECTANGLE	CIRCLE
XPOLY(1,I)	X-coordinate of lowest vertex	X-coordinate of lowest vertex	X-coordinate of center
XPOLY(1,I)	Y-coordinate of lowest vertex	Y-coordinate of lowest vertex	X-coordinate of center
XPOLY(2,I)	X-coordinate of middle vertex	X-coordinate of left-most vertex	0.0
YPOLY(2,I)	Y-coordinate of middle vertex	Y-coordinate of left-most vertex	0.0
XPOLY(3,I)	X-coordinate of highest vertex	Length of rectangle	Radius of circle
YPOLY(3,I)	Y-coordinate of highest vertex	0.0	0.0
XPOLY(4,I)	0.0	0.0	0.0
YPOLY(4,I)	0.0	0.0	0.0
XPOLY(5,I)	0.0	0.0	0.0
YPOLY(5,I)	0.0	0.0	0.0

Note that array elements which do not contain data remain initialized with the floating point value of zero.

The data in the arrays comprising the Vegetation Polygon Table remain constant throughout the simulation. The following subroutines reference, but never modify, the data stored in the above arrays:

LINOBS

LOSINP

MICSOL.